

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-084180

(43)Date of publication of application : 30.03.2001

(51)Int.Cl. G06F 12/16

G06F 12/00

(21)Application number : 11-256019

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 09.09.1999

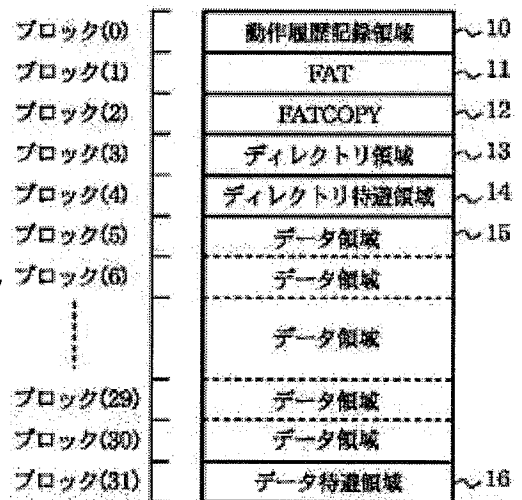
(72)Inventor : NAGAOKA HIDETADA

(54) FILE MANAGING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a file managing device normally resuming a file operation without destroying any file or data even when an abnormal state such as power source interruption is generated during the file operation.

SOLUTION: In this file managing device, an identification code indicating the processing contents of wiring processing or the like are recorded in an operation history recording region 10, and when the writing processing or the like is normally ended, the identification code is deleted. Afterwards, when the recording of the identification code is left in the operation history recording region 10, the identification code is analyzed, and the data stored in the data region or the like are restored.



*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1]While searching a vacant sector of a data area of the flash memory with reference to management data stored in a management domain of a flash memory and writing in file data to the vacant sector, When a writing means which updates the management data, and the above-mentioned writing means perform writing processing or an update process, If an identification code which shows the contents of processing is recorded on an operation history recording region of the flash memory concerned and the writing processing or update process carries out normal termination, A file management device provided with a restoration means to restore data which analyzes the identification code and is stored in the data area and management domain when record of an identification code is left behind to an operation history recording device which eliminates the identification code, and an operation history recording region of the above-mentioned flash memory.

[Claim 2]While searching a vacant sector of a data area of the flash memory with reference to management data stored in a management domain of a flash memory and writing in file data to the vacant sector, When a writing means which updates the management data, and the above-mentioned writing means perform writing processing or an update process, If an identification code which shows the contents of processing is recorded on an operation history recording region of the flash memory concerned and the writing processing or update process carries out normal termination, An operation history recording device which records an identification code which shows normal termination on the operation history recording region concerned, A file management device provided with a restoration means to restore data which analyzes an identification code which shows the contents of processing, and is stored in the data area and management domain when an identification code which shows normal termination is not recorded on an operation history recording region of the above-mentioned flash memory.

[Claim 3]The file management device according to claim 1 or 2, wherein a restoration means performs a restoration process according to an analysis result of an identification code and returns data stored in a data area and a management domain to the contents before writing processing or an update process is performed.

[Claim 4]While searching a block of a data area containing a sector for [the] deletion with reference to management data stored in a management domain of a flash memory and deleting a sector for deletion from the block, When a deleting means which updates the management data, and the above-mentioned deleting means perform deletion or an update process, If an identification code which shows the contents of processing is recorded on an operation history recording region of the flash memory concerned and the deletion or update process carries out normal termination, A file management device provided with a restoration means to restore data which analyzes the identification code and is stored in the data area and management domain when record of an

identification code is left behind to an operation history recording device which eliminates the identification code, and an operation history recording region of the above-mentioned flash memory. [Claim 5]While searching a block of a data area containing a sector for [the] deletion with reference to management data stored in a management domain of a flash memory and deleting a sector for deletion from the block, When a deleting means which updates the management data, and the above-mentioned deleting means perform deletion or an update process, If an identification code which shows the contents of processing is recorded on an operation history recording region of the flash memory concerned and the deletion or update process carries out normal termination, An operation history recording device which records an identification code which shows normal termination on the operation history recording region concerned, A file management device provided with a restoration means to restore data which analyzes an identification code which shows the contents of processing, and is stored in the data area and management domain when an identification code which shows normal termination is not recorded on an operation history recording region of the above-mentioned flash memory.

[Claim 6]The file management device according to claim 4 or 5, wherein a restoration means performs a restoration process according to an analysis result of an identification code and returns data stored in a data area and a management domain to the contents before deletion or an update process is performed.

[Claim 7]When it records the same identification code on it twice in succession [an operation history recording device / an operation history recording region], A file management device given [of Claim 1 restricting a restoration means when the same identification code is recorded twice in succession, and dealing with the identification code noting that it is an effective identification code to Claim 6] in any 1 clause.

[Claim 8]If a detection means to detect a hardware error of a flash memory, and the above-mentioned detection means detect a hardware error, A file management device given [of Claim 1 establishing a control means which stops processing of a writing means and makes processing of a restoration means start to Claim 3] in any 1 clause.

[Claim 9]If a detection means to detect a hardware error of a flash memory, and the above-mentioned detection means detect a hardware error, A file management device given [of Claim 4 establishing a control means which stops processing of a deleting means and makes processing of a restoration means start to Claim 6] in any 1 clause.

[Claim 10]The file management device according to claim 8 or 9 when a detection means cannot receive a status signal of a hardware error from a flash memory beyond in fixed time, wherein a control means performs the same processing as a case where the above-mentioned detection means detects a hardware error.

[Claim 11]A file management device given [of Claim 1 establishing a reading means which reads data from a bank of another side while a writing means is writing in file data to one bank, when a flash memory comprises two banks to Claim 3] in any 1 clause.

[Claim 12]A file management device given [of Claim 4 establishing a reading means which reads data from a bank of another side when a flash memory comprises two banks and a deleting means has deleted a sector from one bank to Claim 6] in any 1 clause.

[Translation done.]

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention relates to the file management device which accesses the flash memory which can eliminate the data of a block unit.

[0002]

[Description of the Prior Art]In recent years, many flash memories are used for the storage of portable information machines and equipment, such as a digital still camera and PDA. A flash memory is a nonvolatile memory which can be written electrically.

Unlike SRAM or DRAM, after power supply cutoff has the feature which can hold data.

[0003]However, the flash memory once needs to eliminate the data, in order are impossible in over-writing and to rewrite the data of the arbitrary addresses in a physical address space. A flash memory usually comprises two or more blocks, and elimination of data can be performed only by a block unit. Although the size of a block changes with chips, they are several kilobytes – hundreds of K bytes.

[0004]In [drawing 19 is an explanatory view showing the logical format on the flash memory in the conventional file management device shown in JP,H11-53248,A, and] a figure, 100 is a reserved area and arbitrary fields [in / 101 / 103 can set FAT (File AllocationTable) and 102 to a directory region, can be set to a data area, and / in 104 / the data area 103].

[0005]As the storage area of a flash memory is shown in drawing 19, it is classified into the reserved area 100, FAT101, the directory region 102, and the data area 103, and a data area comprises two or more blocks. Although the technology of managing a file by FAT and a directory is publicly known technology well known from the former, it describes the easy explanation for below.

[0006]The data area 103 is a field which memorizes file data.

It is logically divided in the field of the fixed size called a sector.

Although access of the data area 103 is performed per sector (the thing treating a mass storage area is accessed in the unit called the cluster which consists of two or more continuous sectors), sector sizes are usually values, such as 256 bytes or 512 bytes.

[0007]A file is stored in two or more sectors with the data size. Even when a file size cannot divide among a sector size, the file occupies a final sector and the data of other files is not stored in the field in which the sector remained. FAT101 has an entry corresponding to each sector at 1 to 1. It can be known from the entry of FAT101 whether data is written in the sector, or it is a vacant sector.

[0008]A file name, a file size, file attribute information, the start sector number of a file, etc. are

recorded on the directory region 102. FAT101 includes the map information which shows in what kind of turn the data of a file is stored in which sector.

[0009]When a file cannot store in one sector, the number of the sector which stores data succeedingly is recorded on the entry of FAT101. When reading file data, the file data currently scatteringly recorded on two or more sectors can be read as one data by reading the entry of FAT sequentially from the start sector number currently recorded on the directory.

[0010]Next, operation in case the conventional file management device updates the file currently recorded on the field 104 is explained. Drawing 20 is a flow chart which shows the contents of processing of the conventional file management device. First, a file management device searches the empty block in the data area 103 (step ST201).

[0011]Next, the inside of the file by which the file management device is recorded on the block (5) with which the field 104 exists, While copying the file currently recorded on fields other than field 104 to an empty block (step ST202), the file after renewal of the file currently recorded on the field 104 is copied to the empty block concerned (step ST203).

[0012]And if the copy of a file is completed, a file management device will eliminate a block (5) (step ST204), and will update the data of FAT101 and the directory region 102 (step ST205).

[0013]

[Problem to be solved by the invention]Since the conventional file management device is constituted as mentioned above, if a flash memory is carried in portable information machines and equipment etc., fault as shown below will occur. That is, since portable information machines and equipment use a cell (battery) for a main power supply, a power supply may be suddenly intercepted by consumption of a cell (or also when a power supply is intercepted by a user's operation mistake, it thinks). For example, if interception of a power supply occurs after execution of step ST204 of drawing 20, in spite of eliminating the block (5), Since the data of FAT101 and the directory region 102 was not updated, when the power supply was rebooted and reading of the applicable file after updating was tried, SUBJECT into which the file (file of a block (5)) of an erasing state will be read occurred.

[0014]Although a block which recorded a file after updating also recorded a file after updating, since it had been recognized to be a vacant sector field from an entry of FAT101, it had SUBJECT which cannot write in a file correctly. When updating FAT101, and FAT101 was eliminated and a power supply was intercepted, data of FAT101 was lost thoroughly and SUBJECT it becomes impossible to read no files occurred.

[0015]It aims at obtaining a file management device which can resume a file operation normally, without destroying a file and data, even if it was made in order that this invention might solve above SUBJECT, and abnormal conditions, such as power supply cutoff, occur during a file operation.

[0016]

[Means for solving problem]If a file management device concerning this invention records an identification code which shows those contents of processing on an operation history recording region of a flash memory and that writing processing or update process carries out normal termination when a writing means performs writing processing or an update process, When an operation history recording device which eliminates the identification code is established and record of an identification code is left behind to an operation history recording region of the flash memory, the identification code is analyzed and data stored in the data area and management domain is restored.

[0017]If the file management device concerning this invention records the identification code which shows those contents of processing on the operation history recording region of a flash memory and that writing processing or update process carries out normal termination when a writing means performs writing processing or an update process, The operation history recording device which records the identification code which shows normal termination on an operation history recording region is established, When the identification code which shows normal termination is not recorded on

the operation history recording region of the flash memory, the identification code which shows the contents of processing is analyzed, and the data stored in the data area and management domain is restored.

[0018]The file management device concerning this invention performs the restoration process according to the analysis result of the identification code, and returns the data stored in the data area and the management domain to the contents before writing processing or an update process is performed.

[0019]If the file management device concerning this invention records the identification code which shows those contents of processing on the operation history recording region of a flash memory and that deletion or update process carries out normal termination when a deleting means performs deletion or an update process, When the operation history recording device which eliminates the identification code is established and record of the identification code is left behind to the operation history recording region of the flash memory, the identification code is analyzed and the data stored in the data area and management domain is restored.

[0020]If a file management device concerning this invention records an identification code which shows those contents of processing on an operation history recording region of a flash memory and that deletion or update process carries out normal termination when a deleting means performs deletion or an update process, An operation history recording device which records an identification code which shows normal termination on an operation history recording region is established, When an identification code which shows normal termination is not recorded on an operation history recording region of the flash memory, an identification code which shows the contents of processing is analyzed, and data stored in the data area and management domain is restored.

[0021]A file management device concerning this invention performs a restoration process according to an analysis result of an identification code, and returns data stored in a data area and a management domain to the contents before deletion or an update process is performed.

[0022]A file management device concerning this invention is restricted when the same identification code is recorded twice in succession, and that identification code is dealt with noting that it is an effective identification code.

[0023]When a detection means detects a hardware error, a file management device concerning this invention stops processing of a writing means, and it is made to make processing of a restoration means start.

[0024]When a detection means detects a hardware error, the file management device concerning this invention stops processing of a deleting means, and it is made to make processing of a restoration means start.

[0025]The file management device concerning this invention is made to perform the same processing as the case where that detection means detects a hardware error, when a detection means cannot receive the status signal of a hardware error from a flash memory beyond in fixed time.

[0026]The file management device concerning this invention establishes the reading means which reads data from the bank of another side, while the writing means is writing in the file data to one bank.

[0027]The file management device concerning this invention establishes the reading means which reads data from the bank of another side, when the deleting means has deleted the sector from one bank.

[0028]

[Mode for carrying out the invention]Hereafter, one form of implementation of this invention is explained.

In [embodiment 1. drawing 1 is a block diagram showing the file management device by this embodiment of the invention 1, and] a figure, 1 saves as a file the data inputted from the outside via

the I/F part 4 at the flash memory 5 according to the input signal from the key input section 3, or, Read the file saved at the flash memory 5, and output data outside via the I/F part 4, or, It is MPU which transmits the file data saved at the flash memory 5 to VRAM6, and displays on the display 7, or deletes the file saved at the flash memory 5.

[0029]While a key input section in which various keys which receive various operations are provided, and 4 input the exterior to data from a switch with which ROM which stores a program etc. whose 2 MPU1 performs, and 3 turn a power supply on and off, or a user, VRAM stored when an I/F part which outputs data outside, a flash memory which 5 can eliminate [of data of a block unit], and 6 display data etc., a display as which seven display data etc., and 8 are RAM used as a work memory, when MPU1 performs various processing. MPU1 constitutes a writing means, a deleting means, an operation history recording device, and a restoration means.

[0030]In [drawing 2 is an explanatory view showing a logical format on the flash memory 5 in a file management device, and] a figure, 10 — as for a directory region and 14, FAT and 12 are [a data area and 16] data shunting fields a directory shunting field and 15 FATCOPY and 13 an operation history recording region and 11. A management domain comprises FAT11 and the directory region 13.

[0031]As a storage area of the flash memory 5 is shown in drawing 2, it is classified into the operation history recording region 10, FAT11, FATCOPY12, the directory region 13, the directory shunting field 14, the data area 15, and the data shunting field 16, and a data area comprises two or more blocks. The flash memory 5 which a file management device in this Embodiment 1 uses assumes that it has 32 blocks.

[0032]As stated previously, a block is an elimination unit of a flash memory. In the operation history recording region 10, a block (0) FAT11 a block (1), FATCOPY12 — a block (2) — the directory region 13 — a block (3) — block (5) – (30) is assigned to the data area 15, and a block (31) is assigned to the directory shunting field 14 for a block (4) in the data shunting field 16.

[0033]128 sectors per block are arranged in the data area 15, and each sector is logically mapped by numbers from 0 to 3327. FAT11 has 2 bytes of entry to one sector, and entry data of an initial state (state where no files exist) is set to '0000'. When a file settled in one sector is recorded, 'FFFF' is recorded on a FAT entry of a corresponding sector, for example. When a file is recorded on two or more sectors, a number of a sector connected with the next is recorded on a FAT entry. 'FFFF' is recorded on a final sector of a file. Search of a vacant sector can be performed, when FAT11 is scanned and entry data searches a thing of '0000'.

[0034]Drawing 3 is an explanatory view showing the format of the directory region 13, and the data for identifying whether 20 is directory checkbite and is a directory [finishing / file registration] or it is an intact directory is recorded in a figure. In an initial state, the directory checkbite 20 presupposes that it is '00'. An intact directory can be searched if the directory checkbite 20 is scanned. The start sector number 21 indicates the number of the sector of the head where a file attribute records a file name and 22, as for 23, the date is recorded, and, as for 24, the data of the applicable file is recorded to be, and 25 are file sizes.

[0035]The operation history recording regions 10 are the only data (henceforth an identification code of operation) in which a processing lapse situation when performing writing processing of a file is shown, and an area for recording the information which is needed in order to restore a file management device at the time of an abnormal occurrence. Usually, it is an erasing state at the time. The timing which records the concrete contents of record and operation history is mentioned later.

[0036]As for the flash memory 5 which a file management device uses, all the bits are set to '1' at the time of elimination. The directory shunting field 14 is a temporary field used when rewriting the contents of the directory region 13. The data shunting field 16 is a temporary field used when rewriting the contents of the data area 15. FATCOPY12 is a field where the same data as FAT11 is recorded. The technique of providing FATCOPY12 is often used from the former, in order to improve

the reliability of a file management device.

[0037]Next, operation is explained. Drawing 4 is a flow chart which shows writing processing of a file by a file management device. First, MPU1 will calculate a sector number required since the file is saved, if a write command of a file is received from the key input section 3. And MPU1 scans FAT11 and it searches a vacant sector by a required sector (step ST1). When there is no vacant sector for a required sector in that case, it is reported that writing of a file is improper (step ST17).

[0038]MPU1 receives the operation history recording region 10, if a vacant sector is searched by a required sector, The number of the identification code 'A000' of operation which shows that it is [file] under writing, and the sector which records the file is recorded (step ST2), and the file is recorded on a corresponding sector (step ST3).

[0039]And MPU1 scans the directory checkbite 20 of the directory region 13, it searches an intact directory (step ST4), and copies the management data of the directory region 13 to the directory shunting field 14 (step ST5). MPU1 adds the identification code 'A001' of operation which shows that it is [directory] under updating to the operation history recording region 10 (step ST6).

[0040]MPU1 will carry out block deletion of the directory region 13, if an identification code 'A001' of operation is added to the operation history recording region 10 (step ST7). And MPU1 merges new directory data and the existing directory data, and it records the management data after merging into the directory region 13 (step ST8).

[0041]Next, MPU1 adds the identification code 'A002' of operation which shows that it is [FAT] under updating to the operation history recording region 10 (step ST9). MPU1 will carry out block deletion of FAT11, if an identification code 'A002' of operation is added to the operation history recording region 10 (step ST10). And MPU1 performs record of a new FAT entry, and write return of an existing FAT entry (step ST11). That is, the existing FAT entry copied to FATCOPY12 and a new FAT entry are merged beforehand, and a FAT entry is updated.

[0042]Next, MPU1 will add an identification code 'A003' of operation which shows that renewal of FAT was completed to the operation history recording region 10, if renewal of FAT11 is completed (step ST12). MPU1 will carry out block deletion of FATCOPY12, if an identification code 'A003' of operation is added to the operation history recording region 10 (step ST13).

[0043]And MPU1 copies management data (FAT entry) of FAT11 to FATCOPY12 in order to prepare for next file writing processing (step ST14). MPU1 carries out block deletion of the directory shunting field 14 (step ST15), and it carries out block deletion of the operation history recording region 10 (step ST16).

[0044]Here, a case where a power supply was intercepted suddenly and carries out abnormal termination in the middle of the above-mentioned file writing processing is considered. Here, it explains as an example as that by which a power supply was intercepted during processing of step ST8 of drawing 4. However, in order to clarify an effect of this invention, a case where a restoration process is not carried out at all is explained first.

[0045]It reboots, and again, if writing processing of a file is tried, the following situations will occur. First, in step ST1, although FAT11 is scanned and a vacant sector is searched, since writing to a FAT entry is not completed at the abnormal termination time, a sector file written in is recognized to be a vacant sector, and overwrite processing of a file is performed in step ST3.

[0046]However, even if the flash memory 5 performs rewriting of data to a field which wrote in data once, without eliminating, it cannot record data correctly. Therefore, although processing advances as what was written in normally in a sequence top, a file of mistaken contents will be saved at the flash memory 5.

[0047]Next, a case where a restoration process by an operation history is performed is explained. Drawing 5 is a flow chart which shows a restoration process performed at the time of a reboot. First, MPU1 checks data of the operation history recording region 10 at the time of a reboot (step ST20).

[0048]When record of an identification code of operation is not left behind to the operation history recording region 10, MPU1 judges it as what an abnormal termination state has not generated (when an identification code of operation is eliminated), ends a restoration process, and shifts to the usual processing. On the other hand, when record of an identification code of operation is left behind, it is judged as what an abnormal termination state generated (when an identification code of operation is not eliminated), and processing after step ST21 is performed.

[0049]If generating of an abnormal termination state is recognized, MPU1 will judge whether abnormal termination was carried out, while analyzing an identification code of operation currently recorded on the operation history recording region 10 at the end and performing which processing among file writing processings (step ST21). In this Embodiment 1, since it assumes that it is what abnormal termination generated at the time of execution of step ST8, an identification code of operation currently recorded on the last of the operation history recording region 10 is that of 'A001'.

[0050]Processing shown by generating time of abnormal termination after step ST22 although the contents of the restoration process differ a little is a flow chart when an identification code 'A001' of operation is detected. When an identification code of operation currently recorded at the end is another code, it shifts to processing of step ST29, step ST30, and step ST31.

[0051]In step ST21, when an identification code 'A001' of operation is detected, MPU1 carries out block deletion of the directory region 13 (step ST22), and it copies management data of the directory shunting field 14 to the directory region 13 (step ST23).

[0052]MPU1 copies data of a block containing a sector which the sector number shows to the data shunting field 16 with reference to a sector number currently recorded on the operation history recording region 10 (step ST24). And MPU1 carries out block deletion of the block containing the sector (step ST25), and it performs processing which returns valid data other than the sector (step ST26). That is, processing which returns only data stored in sectors other than a sector which the sector number shows among sectors which constitute the block to a sector of the original block is performed.

[0053]If MPU1 is returned to a state before writing processing of data is performed in the contents of the data area 15 as mentioned above, it will carry out block deletion of the data shunting field 16 (step ST27), and will carry out block deletion of the operation history recording region 10 (step ST28). Management data of a directory which data of a sector which a FAT entry indicates to be a vacant sector by the above-mentioned restoration process was in an erasing state, and was recorded halfway is also eliminated.

[0054]As mentioned above, an identification code of operation is recorded one by one during writing processing of a file in the operation history recording region 10. Since an identification code of operation is analyzed at the time of a power supply reboot and he is trying to restore the contents of the flash memory 5 if a number of a sector which records a file is also saved in the operation history recording region 10 and abnormal termination generates it by power supply cutoff etc., An effect that a file management device which uses for a storage the flash memory 5 whose reliability of data it is strong in the sudden abnormal termination state, such as sudden power supply cutoff, and is high can be obtained is done so.

[0055]Drawing 6 is a flow chart which shows deletion of a file by a file management device. First, MPU1 will search a number of a sector in which a file to delete is stored, if deletion instructions of a file are received from the key input section 3 (step ST40). And MPU1 will copy data of a block containing the sector to delete to the data shunting field 16, if a sector number is searched (step ST41). MPU1 records an identification code 'A100' of operation which shows that it is during file deletion to the operation history recording region 10, a file number of the file, and a number of a block containing a sector to delete (step ST42).

[0056]And MPU1 performs processing which carries out block deletion of the block containing a

sector to delete (step ST43), and returns valid data other than a file to delete (step ST44). That is, processing which returns only data stored in sectors other than a sector deleted among sectors which constitute the block to a sector of the original block is performed.

[0057]Next, if data writes MPU1 and return processing is performed, it will add an identification code 'A101' of operation which shows that data deletion was completed to the operation history recording region 10 (step ST45). MPU1 will copy management data of the directory region 13 to the directory shunting field 14, if an identification code 'A101' of operation is added to the operation history recording region 10 (step ST46).

[0058]Next, MPU1 adds an identification code 'A102' of operation which shows that it is [directory] under updating to the operation history recording region 10 (step ST47). MPU1 will carry out block deletion of the directory region 13, if an identification code 'A102' of operation is added to the operation history recording region 10 (step ST48). MPU1 makes a directory of an eliminated file an unused state, and it records the newest management data on the directory region 13 (step ST49). Management data concerning files other than a deletion object file copies management data copied to the directory shunting field 14 to the directory region 13.

[0059]Next, MPU1 adds the identification code 'A103' of operation which shows that it is [FAT] under updating to the operation history recording region 10 (step ST50). MPU1 will carry out block deletion of FAT11, if an identification code 'A103' of operation is added to the operation history recording region 10 (step ST51). MPU1 performs record of a new FAT entry, and write return of an existing FAT entry (step ST52). That is, the existing FAT entry copied to FATCOPY12 and a new FAT entry are merged beforehand, and a FAT entry is updated.

[0060]Next, MPU1 will add the identification code 'A104' of operation which shows that renewal of FAT was completed to the operation history recording region 10, if renewal of FAT11 is completed (step ST53). MPU1 will carry out block deletion of FATCOPY12, if an identification code 'A104' of operation is added to the operation history recording region 10 (step ST54).

[0061]And MPU1 copies the management data (FAT entry) of FAT11 to FATCOPY12 in order to prepare for next file deletion (step ST55). MPU1 carries out block deletion of the directory shunting field 14 (step ST57), and it carries out block deletion of the operation history recording region 10 while it carries out block deletion of the data shunting field 16 (step ST56) (step ST58).

[0062]Here, the case where the power supply was intercepted suddenly and carries out abnormal termination in the middle of above-mentioned file deletion is considered. Here, it explains as an example as that by which the power supply was intercepted after processing of step ST43 of drawing 6. Since all the files stored in the block containing the sector for deletion are eliminated, when it reboots without carrying out a restoration process at all and read-out of the file stored in the block is tried, the file of an erasing state will be read.

[0063]Next, the case where the restoration process by an operation history is performed is explained. Drawing 7 is a flow chart which shows the restoration process performed at the time of a reboot. First, MPU1 checks the data of the operation history recording region 10 at the time of a reboot (step ST60).

[0064]When record of the identification code of operation is not left behind to the operation history recording region 10, MPU1 judges it as what the abnormal termination state has not generated (when the identification code of operation is eliminated), ends a restoration process, and shifts to the usual processing. On the other hand, when record of the identification code of operation is left behind, it is judged as what the abnormal termination state generated (when the identification code of operation is not eliminated), and processing after step ST61 is performed.

[0065]If generating of an abnormal termination state is recognized, MPU1 will judge whether abnormal termination was carried out, while analyzing the identification code of operation currently recorded on the operation history recording region 10 at the end and performing which processing among file

deletions (step ST61). In this Embodiment 1, since it assumes that it is what abnormal termination generated after execution of step ST43, the identification code of operation currently recorded on the last of the operation history recording region 10 is 'A100'.

[0066]The processing shown by the generating time of abnormal termination after step ST62 although the contents of the restoration process differ a little is a flow chart when an identification code 'A100' of operation is detected. When the identification code of operation currently recorded at the end is another code, it shifts to processing of step ST67, step ST68, step ST69, and step ST70.

[0067]In step ST61, MPU1 carries out block deletion of the block containing the sector which the sector number shows with reference to the sector number currently recorded on the operation history recording region 10, when an identification code 'A100' of operation is detected (step ST62). And MPU1 copies the data copied to the data shunting field 16 to the above-mentioned block in step ST41 of drawing 6 (step ST63).

[0068]If MPU1 is returned to the state before deletion of data is performed in the contents of the data area 15 as mentioned above, While carrying out block deletion of the data shunting field 16 (step ST64), block deletion of the operation history recording region 10 is carried out (step ST65), and file deletion is rerun (step ST66). Since the number of the file to delete is recorded on the operation history recording region 10, it can recognize by referring to the operation history recording region 10. The eliminated effective file revives by the above-mentioned restoration process.

[0069]As mentioned above, according to this Embodiment 1, an identification code of operation is recorded one by one during file deletion in the operation history recording region 10, If the number of the file to delete and the number of the block containing the sector in which the file to delete is stored are also saved in the operation history recording region 10 and abnormal termination generates them by power supply cutoff etc., Since an identification code of operation is analyzed at the time of a power supply reboot and he is trying to restore the contents of the flash memory 5, the effect that the file management device which uses for a storage the flash memory 5 whose reliability of data it is strong in the sudden abnormal termination state, such as sudden power supply cutoff, and is high can be obtained is done so.

[0070]By the embodiment 2. above-mentioned embodiment 1, when file writing processing or file deletion carried out normal termination, what carries out block deletion of the operation history recording region 10 was shown, but. As the identification code of operation which shows that file writing processing or file deletion carried out normal termination is added to the operation history recording region 10, when the free space of the operation history recording region 10 becomes below certain size, it may be made to carry out block deletion of the operation history recording region 10. The composition of a file management device and the logical format on the flash memory 5 are the same as that of drawing 1 and drawing 2 in the above-mentioned Embodiment 1.

[0071]Next, operation is explained. Drawing 8 is a flow chart which shows writing processing of a file by a file management device. First, MPU1 will calculate a sector number required since the file is saved, if a write command of a file is received from the key input section 3. And MPU1 scans FAT11 and it searches a vacant sector by a required sector (step ST80). When there is no vacant sector for a required sector in that case, it is reported that writing of a file is improper (step ST98).

[0072]MPU1 will check free space of the operation history recording region 10, if a vacant sector is searched by a required sector (step ST81). And MPU1 carries out block deletion of the operation history recording region 10, when free space is below certain size (step ST82). On the other hand, when free space is more than certain size, the operation history recording region 10 is received, A number of an identification code 'A000' of operation which shows that it is [file] under writing, and a sector which records the file is recorded (step ST83), and the file is recorded on a corresponding sector (step ST84).

[0073]And MPU1 scans the directory checkbite 20 of the directory region 13, it searches an intact

directory (step ST85), and copies management data of the directory region 13 to the directory shunting field 14 (step ST86). MPU1 adds an identification code 'A001' of operation which shows that it is [directory] under updating to the operation history recording region 10 (step ST87).

[0074]MPU1 will carry out block deletion of the directory region 13, if an identification code 'A001' of operation is added to the operation history recording region 10 (step ST88). And MPU1 merges new directory data and the existing directory data, and it records management data after merging into the directory region 13 (step ST89).

[0075]Next, MPU1 adds an identification code 'A002' of operation which shows that it is [FAT] under updating to the operation history recording region 10 (step ST90). MPU1 will carry out block deletion of FAT11, if an identification code 'A002' of operation is added to the operation history recording region 10 (step ST91). And MPU1 performs record of a new FAT entry, and write return of an existing FAT entry (step ST92). That is, an existing FAT entry copied to FATCOPY12 and a new FAT entry are merged beforehand, and a FAT entry is updated.

[0076]Next, MPU1 will add the identification code 'A003' of operation which shows that renewal of FAT was completed to the operation history recording region 10, if renewal of FAT11 is completed (step ST93). MPU1 will carry out block deletion of FATCOPY12, if an identification code 'A003' of operation is added to the operation history recording region 10 (step ST94).

[0077]And MPU1 copies the management data (FAT entry) of FAT11 to FATCOPY12 in order to prepare for next file writing processing (step ST95). MPU1 carries out block deletion of the directory shunting field 14 (step ST96), and it adds the identification code 'A0FF' of operation which shows that file writing processing carried out normal termination to the operation history recording region 10 (step ST97).

[0078]In the above-mentioned Embodiment 1, when the operation history recording region 10 was not eliminated at the time of power supply starting, analyzed the identification code of operation, and were performing the restoration process, but. According to this Embodiment 2, when there is no identification code of operation recorded on the operation history recording region 10 at the end at the identification code of operation which shows that normal termination was carried out, that identification code of operation is analyzed and it is made to perform the same restoration process as the above-mentioned Embodiment 1.

[0079]About file deletion, when there is no identification code of operation recorded on the operation history recording region 10 at the end like the above-mentioned file writing processing at an identification code of operation which shows that normal termination was carried out, the identification code of operation is analyzed and it is made to perform the same restoration process as the above-mentioned Embodiment 1.

[0080]It has the fault that the flash memory 5 has writing of data, and long time which block deletion takes compared with reading time of data. As mentioned above, when file writing processing or file deletion carries out normal termination according to this Embodiment 2. Since it will be made to carry out block deletion if an identification code of operation which shows that normal termination was carried out is added and free space of the operation history recording region 10 becomes below certain size, An effect that can reduce overheads of blanking time now and a throughput of a system improves as a result is done so.

[0081]Although the embodiment 3. above-mentioned embodiments 1 and 2 showed what records an identification code of operation on the operation history recording region 10 one by one during file writing processing or file deletion, As shown in drawing 12, when recording an identification code of operation one by one, it may be made to record the identification code of operation twice in succession to the operation history recording region 10. Composition of a file management device and a logical format on the flash memory 5 are the same as that of drawing 1 and drawing 2 in the above-mentioned Embodiment 1.

[0082]By the above-mentioned Embodiment 1, as an example of the abnormal termination under file writing processing, while performing processing of step ST8 of drawing 4, the case where a power supply was intercepted was explained. However, a power supply may be intercepted in the midst of recording the identification code of operation as a rare case. If a power supply is intercepted during operation history record, an identification code of operation may not be recorded correctly and a restoration process may not be performed correctly.

[0083]Drawing 9 is an explanatory view showing the recorded state of the operation history before performing step ST3, when a sector number writes a file in the sector of '0010', '0011', '0012', '0013', and '0014'. '0010', '0011', '0012', '0013', and '0014' which continue after 'A000' are a number of the sector which records a file. 'FFFF' shows that data is an erasing state.

[0084]On the other hand, in step ST2, while drawing 10 is recording a number of a sector which records a file on the operation history recording region 10, it shows an example of a recorded state of an operation history at the time of carrying out abnormal termination. The state where data which should be essentially recorded as '0013' has been accidentally recorded as '0000' is shown. In this case, since an identification code of operation detected is 'A000', restoration process ** (step ST29) of drawing 5 is performed. Drawing 11 is a flow chart which shows the contents of processing of restoration process **.

[0085]MPU1 copies the data of the block containing the sector which the sector number shows to the data shunting field 16 with reference to the sector number currently recorded on the operation history recording region 10 (step ST100). And MPU1 carries out block deletion of the block containing the sector (step ST101), and it performs processing which returns valid data other than the sector (step ST102). That is, processing which returns only the data stored in sectors other than the sector which the sector number shows among the sectors which constitute the block to the sector of the original block is performed. And MPU1 carries out block deletion of the data shunting field 16 (step ST103), and it carries out block deletion of the operation history recording region 10 (step ST104).

[0086]As mentioned above, since a restoration process performs processing for eliminating the data stored in the sector of the sector number currently recorded on the operation history recording region 10, When a restoration process is performed by the operation history recorded state shown in drawing 10, the data stored in the sector of a sector number '0000' will be eliminated accidentally.

[0087]Like drawing 10, in step ST2, while drawing 12 is recording the number of the sector which records a file on the operation history recording region 10, it shows an example of the recorded state of the operation history at the time of carrying out abnormal termination. The data in which '0013' should be recorded twice in succession essentially serves as '0000' and 'FFFF'.

[0088]According to this Embodiment 3, in step ST100 of drawing 11, the same value treats only the identification code of operation currently recorded twice in succession as valid data. Therefore, '0000' is treated as invalid data and the data of a sector number '0000' is not eliminated accidentally.

[0089]Here, also in file deletion, although operation of file writing processing was explained, since operation is the same, explanation is omitted only by data recorded on an operation history differing.

[0090]As mentioned above, according to this Embodiment 3, an identification code of operation which discriminates a processing lapse situation from a logical address which writes in a file at the time of file writing processing is recorded on the operation history recording region 10 one by one twice in succession, Since he is trying to record an identification code of operation which discriminates a processing lapse situation from a number of a file deleted at the time of file deletion, and a number of a block under elimination one by one on the operation history recording region 10 twice in succession, Even if it carries out abnormal termination during operation history record, an effect that a file management device which uses for a storage the flash memory 5 which could analyze an operation history correctly and can perform a restoration process now, and whose reliability of data improved further as a result can be obtained is done so.

[0091]embodiment 4. -- when an error of hardware status of the flash memory 5 is detected and an error is detected, a file management device of this Embodiment 4 interrupts file writing processing or file deletion, and is made to perform a restoration process.

[0092]Composition of a file management device and a logical format on the flash memory 5 are the same as that of drawing 1 and drawing 2 in the above-mentioned Embodiment 1. However, MPU1 constitutes a detection means and a control means from this Embodiment 4 further.

[0093]Unlike DRAM or SRAM, the flash memory 5 is performed when writing of data and elimination of a block publish a software command from MPU1. Drawing 13 is a flow chart which shows program commands which perform writing processing of data to the flash memory 5.

[0094]First, MPU1 performs processing which writes in a program setup command to the flash memory 5 (step ST110). And MPU1 will perform processing which writes in data to the program address of the flash memory 5, if a program setup command is written in (step ST111).

[0095]MPU1 will check the error status of the flash memory 5, if it will stand by until the status of the flash memory 5 will be in a ready state (step ST112), and the status of the flash memory 5 will be in a ready state (step ST113). Thereby, MPU1 can detect the hardware error of the flash memory 5. Therefore, when the hardware error of the flash memory 5 is detected, subsequent file writing processings are interrupted and it shifts to the restoration process by an operation history.

[0096]Drawing 14 is a flow chart which shows the block erasion command which performs block deletion to the flash memory 5. First, MPU1 performs processing which writes in a block erasion setup command to the flash memory 5 (step ST120). And MPU1 will perform processing which writes in block erasion / check command to the flash memory 5, if a block erasion setup command is written in (step ST121).

[0097]MPU1 will check error status of the flash memory 5, if it will stand by until status of the flash memory 5 will be in a ready state (step ST122), and status of the flash memory 5 will be in a ready state (step ST123). Thereby, MPU1 can detect a hardware error of the flash memory 5. Therefore, when a hardware error of the flash memory 5 is detected, subsequent file deletion is interrupted and it shifts to a restoration process by an operation history.

[0098]As mentioned above, since according to this Embodiment 4 file writing processing or file deletion will be interrupted and it will be made to perform a restoration process if an error of hardware status of the flash memory 5 is detected, An effect that a file management device which uses for a storage the flash memory 5 whose reliability of data improved further can be obtained is done so.

[0099]When an error of hardware status of the flash memory 5 was detected, interrupted file writing processing or file deletion for the embodiment 5. above-mentioned embodiment 4, and what performs a restoration process was shown by it, but. When a command is published to the flash memory 5, a timer which starts Measurement Division is formed and the flash memory 5 to a fixed time response cannot be found, file writing processing or file deletion is interrupted, and it may be made to perform a restoration process. Composition of a file management device and a logical format on the flash memory 5 are the same as that of drawing 1 and drawing 2 in the above-mentioned Embodiment 1.

[0100]In step ST112 in drawing 13 and drawing 14, and ST122, the processing which stands by until status will be in a ready state polls the status read from the flash memory 5, and is checking the specific bit in the status. According to a certain fault, unless the specific bit will be in a ready state, a program may be unable to shift to the next processing from polling processing.

[0101]Drawing 15 is a flow chart which shows the contents of processing of the program commands in this Embodiment 5. First, MPU1 performs processing which writes in a program setup command to the flash memory 5 (step ST130). And MPU1 will perform processing which writes in data to the program address of the flash memory 5, if a program setup command is written in (step ST131). In that case, MPU1 starts a timer (step ST132).

[0102]Like the above-mentioned Embodiment 4, MPU1 stands by until status of the flash memory 5 will be in a ready state, but (step ST133). If a value of waiting [the] and a timer is checked (step ST134) and a value of a timer becomes beyond constant value, status of the flash memory 5 will interrupt file writing processing, be [it / a non-ready state / continue], and will be made to perform a restoration process.

[0103]However, if a waiting state will be continued and status of the flash memory 5 will be in a ready state as long as a value of a timer is below constant value, error status of the flash memory 5 will be checked like the above-mentioned Embodiment 4 (step ST135). When an error is detected, file writing processing is interrupted and a restoration process is performed.

[0104]By this Embodiment 5, when a timer was started during file writing processing and a value of a timer became beyond constant value, what performs a restoration process was shown, but. By file deletion, what is necessary is just to start a timer after processing of step ST121 in drawing 14, and a restoration process as well as file writing processing can be started.

[0105]As mentioned above, when according to this Embodiment 5 a command is published to the flash memory 5, a timer which starts Measurement Division is formed and the flash memory 5 to a fixed time response cannot be found, Since file writing processing or file deletion is interrupted and it is made to perform a restoration process, an effect that a file management device which uses for a storage the flash memory 5 whose reliability of data improved further can be obtained is done so.

[0106]It comprises two banks and there are some flash memories of embodiment 6. recent years which have possible reading data from a bank of another side during writing processing or deletion to one bank. The above-mentioned function is called a dual operation function, a background operation function, etc. The flash memory 5 with the above-mentioned function is used for a file management device of this Embodiment 6, and it plans performance improvement of processing time in file writing processing or file deletion.

[0107]Drawing 16 is an explanatory view showing a logical format on the flash memory 5 in a file management device. The flash memory 5 is classified into a bank (1) and a bank (2), block (0) - (15) is assigned to a bank (1), and block (16) - (31) is assigned to a bank (2). It is possible to read data of a bank (2), while performing writing processing and deletion to a bank (1), and conversely, while performing writing processing and deletion to a bank (2), it is possible to read data of a bank (1).

[0108]In this Embodiment 6, in the operation history recording region 10, a block (0), To FAT11, a block (1) FATCOPY12 a block (29), the directory region 13 -- a block (2) -- block (3) - (28) is assigned to the data area 15, and the block (31) is assigned to the directory shunting field 14 for the block (30) in the data shunting field 16. The composition of a file management device is the same as that of drawing 1 in Embodiment 1. However, MPU1 constitutes the reading means from this Embodiment 6 of this further.

[0109]As the front embodiment described, in file writing processing, file deletion, a restoration process, etc., processing which copies FAT, a directory, a file, etc. to another block from arbitrary blocks is performed frequently. Drawing 17 is an explanatory view explaining the operation at the time of the copy processing execution at the time of using the flash memory which does not have a dual operation function.

[0110]Data of certain size is read from a block of a copied material, data is stored in RAM of RAM8 and MPU1 inside, then, program commands are published to the flash memory 5, and it is data writing **** to another block. Repeat execution of the above-mentioned reading operation and the write operation is carried out, and data for 1 block is copied. Although a write time requires long time compared with read time, MPU1 shifts to reading operation of the following data, after standing by until status which shows that write operation was completed is obtained.

[0111]Drawing 18 is an explanatory view explaining operation at the time of copy processing execution at the time of using a flash memory which has a dual operation function. A block of a part

of FAT11, directory region 13, and data area 15 is constituted by bank (1), and FATCOPY12, the directory shunting field 14, and the data shunting field 16 are constituted by bank (2).

[0112]Therefore, since reading operation and write operation can be overlapped and data of a bank (1) can be performed as shown in drawing 18 when copying data of a bank (2) to a bank (1) on a bank (2) or and conversely, data processing time can be shortened. As mentioned above, although it explained that data processing time was shortened by example with a dual operation function in data copy processing during a block, it is also possible to apply to another processing of performing search of file writing and an intact directory also [except data copy processing] etc.

[0113]As mentioned above, according to this Embodiment 6, the flash memory 5 with a dual operation function is used, Since the block with which a copy of data etc. are performed frequently constitutes the logical format so that it may become a separate bank, data processing time is shortened and the effect that the throughput of a system improves is done so.

[0114]

[Effect of the Invention]As mentioned above, if according to this invention the identification code which shows those contents of processing is recorded on the operation history recording region of a flash memory and that writing processing or update process carries out normal termination when a writing means performs writing processing or an update process, When the operation history recording device which eliminates the identification code is established and record of the identification code is left behind to the operation history recording region of the flash memory, the identification code is analyzed, Since it constituted so that the data stored in the data area and management domain might be restored, There is an effect which can resume a file operation normally without destroying a file and data, even if abnormal conditions, such as power supply cutoff, occur, while the writing means is performing writing processing or an update process.

[0115]If according to this invention the identification code which shows those contents of processing is recorded on the operation history recording region of a flash memory and that writing processing or update process carries out normal termination when a writing means performs writing processing or an update process, The operation history recording device which records the identification code which shows normal termination on an operation history recording region is established, When the identification code which shows normal termination is not recorded on the operation history recording region of the flash memory, the identification code which shows the contents of processing is analyzed, Since it constituted so that the data stored in the data area and management domain might be restored, There is an effect which can resume a file operation normally without destroying a file and data, even if abnormal conditions, such as power supply cutoff, occur, while the writing means is performing writing processing or an update process.

[0116]Since according to this invention it constituted so that the data which performs the restoration process according to the analysis result of the identification code, and is stored in the data area and the management domain might be returned to the contents before writing processing or an update process is performed, It is effective in the flash memory whose reliability of data it is strong in the sudden abnormal termination state, such as sudden power supply cutoff, and is high being obtained.

[0117]If according to this invention the identification code which shows those contents of processing is recorded on the operation history recording region of a flash memory and that deletion or update process carries out normal termination when a deleting means performs deletion or an update process, When the operation history recording device which eliminates the identification code is established and record of the identification code is left behind to the operation history recording region of the flash memory, the identification code is analyzed, Since it constituted so that the data stored in the data area and management domain might be restored, There is an effect which can resume a file operation normally without destroying a file and data, even if abnormal conditions, such as power supply cutoff, occur, while the deleting means is performing deletion or an update process.

[0118]If according to this invention the identification code which shows those contents of processing is recorded on the operation history recording region of a flash memory and that deletion or update process carries out normal termination when a deleting means performs deletion or an update process, The operation history recording device which records the identification code which shows normal termination on an operation history recording region is established, When the identification code which shows normal termination is not recorded on the operation history recording region of the flash memory, the identification code which shows the contents of processing is analyzed, Since it constituted so that the data stored in the data area and management domain might be restored, There is an effect which can resume a file operation normally without destroying a file and data, even if abnormal conditions, such as power supply cutoff, occur, while the deleting means is performing deletion or an update process.

[0119]Since according to this invention it constituted so that the data which performs the restoration process according to the analysis result of the identification code, and is stored in the data area and the management domain might be returned to the contents before deletion or an update process is performed, It is effective in the flash memory whose reliability of data it is strong in the sudden abnormal termination state, such as sudden power supply cutoff, and is high being obtained.

[0120]Since according to this invention it constituted so that it restricted when the same identification code is recorded twice in succession, and that identification code might be dealt with noting that it is an effective identification code, Even if it carries out abnormal termination during operation history record, it is effective in the ability to obtain the flash memory which could analyze the operation history correctly and can perform a restoration process now and whose reliability of data improved further as a result.

[0121]According to this invention, since it constituted so that processing of a writing means might be stopped and processing of a restoration means might be made to start when the detection means detected the hardware error, it is effective in the ability to obtain the flash memory whose reliability of data improved further.

[0122]According to this invention, since it constituted so that processing of a deleting means might be stopped and processing of a restoration means might be made to start when the detection means detected the hardware error, it is effective in the ability to obtain the flash memory whose reliability of data improved further.

[0123]Since according to this invention it constituted so that the same processing as the case where that detection means detects a hardware error might be performed when a detection means was not able to receive the status signal of a hardware error from a flash memory beyond in fixed time, Even when a certain fault occurs in a flash memory and the status signal of a hardware error cannot be received, it is effective in the ability to obtain the flash memory where whose reliability of data a restoration means can perform a restoration process now and improved further as a result.

[0124]According to this invention, since it constituted so that the reading means which reads data from the bank of another side might be established while the writing means was writing in the file data to one bank, data processing time is shortened and it is effective in the throughput of a system improving.

[0125]According to this invention, since it constituted so that the reading means which reads data from the bank of another side might be established when the deleting means had deleted the sector from one bank, data processing time is shortened and it is effective in the throughput of a system improving.

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-84180

(P2001-84180A)

(43) 公開日 平成13年3月30日 (2001.3.30)

(51) Int.Cl. ⁷	識別記号	F I	テ-マ-ト* (参考)
G 0 6 F 12/16	3 4 0	G 0 6 F 12/16	3 4 0 P 5 B 0 1 8
12/00	5 4 2	12/00	5 4 2 M 5 B 0 8 2

審査請求 未請求 請求項の数12 O L (全 16 頁)

(21) 出願番号 特願平11-256019

(22) 出願日 平成11年9月9日 (1999.9.9)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 長岡 秀忠

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74) 代理人 100066474

弁理士 田澤 博昭 (外1名)

Fターム(参考) 5B018 GA04 HA22 KA11 LA06 NA06

QA05 QA06

5B082 DB03 DC06 DD04 EA01

(54) 【発明の名称】 ファイル管理装置

(57) 【要約】 (修正有)

【課題】 例えば、電源の遮断がブロックの消去後に発生すると、ブロックが消去されているにも拘わらず、F A T及びディレクトリ領域のデータが更新されないため、電源を再起動して、更新後の該当ファイルの読み込みを試みると、消去状態のファイル（ブロックのファイル）が読み込まれてしまう課題があった。

【解決手段】 書込処理等の処理内容を示す識別コードを動作履歴記録領域10に記録し、その書込処理等が正常終了すると、その識別コードを消去する。その後、動作履歴記録領域10に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域等に格納されているデータを復旧する。

ブロック(0)	動作履歴記録領域	~ 10
ブロック(1)	FAT	~ 11
ブロック(2)	FATCOPY	~ 12
ブロック(3)	ディレクトリ領域	~ 13
ブロック(4)	ディレクトリ待避領域	~ 14
ブロック(5)	データ領域	~ 15
ブロック(6)	データ領域	
⋮	データ領域	
ブロック(29)	データ領域	
ブロック(30)	データ領域	
ブロック(31)	データ待避領域	~ 16

【特許請求の範囲】

【請求項1】 フラッシュメモリの管理領域に格納されている管理データを参照して、そのフラッシュメモリのデータ領域の空きセクタを検索し、その空きセクタに対してファイルデータを書き込むとともに、その管理データを更新する書込手段と、上記書込手段が書込処理又は更新処理を実行する際、その処理内容を示す識別コードを当該フラッシュメモリの動作履歴記録領域に記録し、その書込処理又は更新処理が正常終了すると、その識別コードを消去する動作履歴記録手段と、上記フラッシュメモリの動作履歴記録領域に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧する復旧手段とを備えたファイル管理装置。

【請求項2】 フラッシュメモリの管理領域に格納されている管理データを参照して、そのフラッシュメモリのデータ領域の空きセクタを検索し、その空きセクタに対してファイルデータを書き込むとともに、その管理データを更新する書込手段と、上記書込手段が書込処理又は更新処理を実行する際、その処理内容を示す識別コードを当該フラッシュメモリの動作履歴記録領域に記録し、その書込処理又は更新処理が正常終了すると、正常終了を示す識別コードを当該動作履歴記録領域に記録する動作履歴記録手段と、上記フラッシュメモリの動作履歴記録領域に正常終了を示す識別コードが記録されていない場合、その処理内容を示す識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧する復旧手段とを備えたファイル管理装置。

【請求項3】 復旧手段は、識別コードの解析結果に応じた復旧処理を実行して、データ領域及び管理領域に格納されているデータを書込処理又は更新処理が実行される前の内容に戻すことを特徴とする請求項1又は請求項2記載のファイル管理装置。

【請求項4】 フラッシュメモリの管理領域に格納されている管理データを参照して、その削除対象のセクタを含むデータ領域のブロックを検索し、そのブロックから削除対象のセクタを削除するとともに、その管理データを更新する削除手段と、上記削除手段が削除処理又は更新処理を実行する際、その処理内容を示す識別コードを当該フラッシュメモリの動作履歴記録領域に記録し、その削除処理又は更新処理が正常終了すると、その識別コードを消去する動作履歴記録手段と、上記フラッシュメモリの動作履歴記録領域に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧する復旧手段とを備えたファイル管理装置。

【請求項5】 フラッシュメモリの管理領域に格納されている管理データを参照して、その削除対象のセクタを含むデータ領域のブロックを検索し、そのブロックから削除対象のセクタを削除するとともに、その管理データ

を更新する削除手段と、上記削除手段が削除処理又は更新処理を実行する際、その処理内容を示す識別コードを当該フラッシュメモリの動作履歴記録領域に記録し、その削除処理又は更新処理が正常終了すると、正常終了を示す識別コードを当該動作履歴記録領域に記録する動作履歴記録手段と、上記フラッシュメモリの動作履歴記録領域に正常終了を示す識別コードが記録されていない場合、その処理内容を示す識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧する復旧手段とを備えたファイル管理装置。

【請求項6】 復旧手段は、識別コードの解析結果に応じた復旧処理を実行して、データ領域及び管理領域に格納されているデータを削除処理又は更新処理が実行される前の内容に戻すことを特徴とする請求項4又は請求項5記載のファイル管理装置。

【請求項7】 動作履歴記録手段が同一の識別コードを動作履歴記録領域に2回続けて記録する場合、復旧手段は同一の識別コードが2回続けて記録されている場合に限り、その識別コードを有効な識別コードであるとして取り扱うことを特徴とする請求項1から請求項6のうちのいずれか1項記載のファイル管理装置。

【請求項8】 フラッシュメモリのハードウェアエラーを検出する検出手段と、上記検出手段がハードウェアエラーを検出すると、書込手段の処理を中止させて復旧手段の処理を開始させる制御手段とを設けたことを特徴とする請求項1から請求項3のうちのいずれか1項記載のファイル管理装置。

【請求項9】 フラッシュメモリのハードウェアエラーを検出する検出手段と、上記検出手段がハードウェアエラーを検出すると、削除手段の処理を中止させて復旧手段の処理を開始させる制御手段とを設けたことを特徴とする請求項4から請求項6のうちのいずれか1項記載のファイル管理装置。

【請求項10】 検出手段がフラッシュメモリからハードウェアエラーのステータス信号を一定時間以上受信できない場合、制御手段は上記検出手段がハードウェアエラーを検出した場合と同様の処理を実行することを特徴とする請求項8又は請求項9記載のファイル管理装置。

【請求項11】 フラッシュメモリが2個のバンクから構成されている場合、書込手段が一方のバンクに対してファイルデータを書き込んでいる時、他方のバンクからデータを読み込む読込手段を設けたことを特徴とする請求項1から請求項3のうちのいずれか1項記載のファイル管理装置。

【請求項12】 フラッシュメモリが2個のバンクから構成されている場合、削除手段が一方のバンクからセクタを削除している時、他方のバンクからデータを読み込む読込手段を設けたことを特徴とする請求項4から請求項6のうちのいずれか1項記載のファイル管理装置。

【発明の詳細な説明】

10

20

30

40

50

【0001】

【発明の属する技術分野】この発明は、ブロック単位のデータの消去が可能なフラッシュメモリをアクセスするファイル管理装置に関するものである。

【0002】

【従来の技術】近年、デジタルスチルカメラやPDAなどの携帯型情報機器の記憶媒体にはフラッシュメモリが多く使用されている。フラッシュメモリは、電氣的に読み書きが可能な不揮発性のメモリであり、SRAMやDRAMと異なり、電源遮断後もデータを保持することができる特徴を有している。

【0003】しかし、フラッシュメモリはオーバーライトが不可能であり、物理アドレス空間における任意のアドレスのデータを書き換えるには、そのデータを一旦消去する必要がある。フラッシュメモリは、通常、複数のブロックから構成され、データの消去はブロック単位でのみ行うことができる。ブロックのサイズはチップによって異なるが、数キロバイト～数百キロバイトである。

【0004】図19は例えば特開平11-53248号公報に示された従来のファイル管理装置におけるフラッシュメモリ上の論理フォーマットを示す説明図であり、図において、100は予約領域、101はFAT (File Allocation Table)、102はディレクトリ領域、103はデータ領域、104はデータ領域103における任意の領域である。

【0005】フラッシュメモリの記憶領域は、図19に示すように、予約領域100、FAT101、ディレクトリ領域102、データ領域103に区分けされていて、データ領域は複数のブロックから構成される。FATとディレクトリでファイルを管理する技術は、従来からよく知られている公知の技術であるが、以下に簡単な説明を記述する。

【0006】データ領域103は、ファイルデータを記憶する領域であり、セクタと呼ばれる固定サイズの領域で論理的に分けられている。データ領域103のアクセスは、セクタ単位で行うが（大容量の記憶領域を扱うものは、連続した複数のセクタからなるクラスタと呼ばれる単位でアクセスを行う）、セクタサイズは通常、256バイトあるいは512バイトといった値である。

【0007】ファイルは、そのデータサイズにより複数のセクタに格納される。ファイルサイズがセクタサイズで割り切れない場合でも、最終セクタはそのファイルが占有し、セクタの余った領域に他のファイルのデータが格納されることはない。FAT101は、それぞれのセクタに1対1で対応したエントリを持っており、セクタにデータが書き込まれているか、空きセクタであるかは、FAT101のエントリから知ることができる。

【0008】ディレクトリ領域102には、ファイル名、ファイルサイズ、ファイル属性情報、ファイルの開始セクタ番号などが記録されている。また、FAT10

1はファイルのデータがどのセクタにどのような順番で格納されているかを示すマップ情報を含んでいる。

【0009】ファイルが1つのセクタに格納しきれない場合、FAT101のエントリにはデータを引き続き格納するセクタの番号が記録される。ファイルデータを読み出すときは、ディレクトリに記録されている開始セクタ番号から順にFATのエントリを読み出すことにより、複数のセクタにばらばらに記録されているファイルデータを1つのデータとして読み出すことができる。

【0010】次に、従来のファイル管理装置が領域104に記録されているファイルを更新する場合の動作を説明する。図20は従来のファイル管理装置の処理内容を示すフローチャートである。まず、ファイル管理装置は、データ領域103における空きブロックを検索する（ステップST201）。

【0011】次に、ファイル管理装置は、領域104が存在するブロック（5）に記録されているファイルのうち、領域104以外の領域に記録されているファイルを空きブロックにコピーするとともに（ステップST202）、その領域104に記録されていたファイルの更新後のファイルを当該空きブロックにコピーする（ステップST203）。

【0012】そして、ファイル管理装置は、ファイルのコピーを完了すると、ブロック（5）を消去し（ステップST204）、FAT101及びディレクトリ領域102のデータを更新する（ステップST205）。

【0013】

【発明が解決しようとする課題】従来のファイル管理装置は以上のように構成されているので、フラッシュメモリを携帯型情報機器などに搭載すると、次に示すような不具合が発生する。即ち、携帯型情報機器は、主電源に電池（バッテリー）を使用するため、電池の消耗によって電源が突然遮断される場合がある（あるいは、ユーザの誤操作によって電源が遮断される場合も考えられる）。例えば、電源の遮断が図20のステップST204の実行後に発生すると、ブロック（5）が消去されているにも拘わらず、FAT101及びディレクトリ領域102のデータが更新されていないため、電源を再起動して、更新後の該当ファイルの読み込みを試みると、消去状態のファイル（ブロック（5）のファイル）が読み込まれてしまう課題があった。

【0014】また、更新後のファイルを記録したブロックも、更新後のファイルを記録したにも拘わらず、FAT101のエントリからは空きセクタ領域であると認識されるので、ファイルを正しく書き込めない課題があった。さらに、FAT101を更新する際、FAT101を消去した時に電源が遮断されると、FAT101のデータが完全に失われ、全てのファイルを読み込めなくなる課題があった。

【0015】この発明は上記のような課題を解決するた

めになされたもので、ファイル操作中に電源遮断などの異常状態が発生しても、ファイルやデータが破壊されることなく、ファイル操作を正常に再開することができるファイル管理装置を得ることを目的とする。

【0016】

【課題を解決するための手段】この発明に係るファイル管理装置は、書込手段が書込処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その書込処理又は更新処理が正常終了すると、その識別コードを消去する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するようにしたものである。

【0017】この発明に係るファイル管理装置は、書込手段が書込処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その書込処理又は更新処理が正常終了すると、正常終了を示す識別コードを動作履歴記録領域に記録する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に正常終了を示す識別コードが記録されていない場合、その処理内容を示す識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するようにしたものである。

【0018】この発明に係るファイル管理装置は、識別コードの解析結果に応じた復旧処理を実行して、データ領域及び管理領域に格納されているデータを書込処理又は更新処理が実行される前の内容に戻すようにしたものである。

【0019】この発明に係るファイル管理装置は、削除手段が削除処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その削除処理又は更新処理が正常終了すると、その識別コードを消去する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するようにしたものである。

【0020】この発明に係るファイル管理装置は、削除手段が削除処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その削除処理又は更新処理が正常終了すると、正常終了を示す識別コードを動作履歴記録領域に記録する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に正常終了を示す識別コードが記録されていない場合、その処理内容を示す識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するようにしたものである。

【0021】この発明に係るファイル管理装置は、識別

コードの解析結果に応じた復旧処理を実行して、データ領域及び管理領域に格納されているデータを削除処理又は更新処理が実行される前の内容に戻すようにしたものである。

【0022】この発明に係るファイル管理装置は、同一の識別コードが2回続けて記録されている場合に限り、その識別コードを有効な識別コードであるとして取り扱うようにしたものである。

【0023】この発明に係るファイル管理装置は、検出手段がハードウェアエラーを検出すると、書込手段の処理を中止させて復旧手段の処理を開始させるようにしたものである。

【0024】この発明に係るファイル管理装置は、検出手段がハードウェアエラーを検出すると、削除手段の処理を中止させて復旧手段の処理を開始させるようにしたものである。

【0025】この発明に係るファイル管理装置は、検出手段がフラッシュメモリからハードウェアエラーのステータス信号を一定時間以上受信できない場合、その検出手段がハードウェアエラーを検出した場合と同様の処理を実行するようにしたものである。

【0026】この発明に係るファイル管理装置は、書込手段が一方のバンクに対してファイルデータを書き込んでいる時、他方のバンクからデータを読み込む読込手段を設けるようにしたものである。

【0027】この発明に係るファイル管理装置は、削除手段が一方のバンクからセクタを削除している時、他方のバンクからデータを読み込む読込手段を設けるようにしたものである。

【0028】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態1. 図1はこの発明の実施の形態1によるファイル管理装置を示す構成図であり、図において、1はキー入力部3からの入力信号に応じて、I/F部4を経由して外部から入力されるデータをフラッシュメモリ5にファイルとして保存したり、フラッシュメモリ5に保存されているファイルを読み出してI/F部4を介してデータを外部に出力したり、フラッシュメモリ5に保存されているファイルデータをVRAM6に転送して表示部7に表示したり、あるいは、フラッシュメモリ5に保存されているファイルを削除したりするMPUである。

【0029】2はMPU1が実行するプログラム等を格納するROM、3は電源をオンオフするスイッチやユーザから各種操作を受け付ける各種キーが設けられているキー入力部、4は外部からデータを入力するとともに、データを外部に出力するI/F部、5はブロック単位のデータの消去が可能なフラッシュメモリ、6はデータ等を表示する際に格納するVRAM、7はデータ等を表示する表示部、8はMPU1が各種処理を実行する際、ワ

ークメモリとして使用されるRAMである。なお、MPU1は書込手段、削除手段、動作履歴記録手段及び復旧手段を構成する。

【0030】図2はファイル管理装置におけるフラッシュメモリ5上の論理フォーマットを示す説明図であり、図において、10は動作履歴記録領域、11はFAT、12はFATCOPY、13はディレクトリ領域、14はディレクトリ待避領域、15はデータ領域、16はデータ待避領域である。なお、FAT11及びディレクトリ領域13から管理領域が構成されている。

【0031】フラッシュメモリ5の記憶領域は、図2に示すように、動作履歴記録領域10、FAT11、FATCOPY12、ディレクトリ領域13、ディレクトリ待避領域14、データ領域15、データ待避領域16に区分けされていて、データ領域は複数のブロックから構成される。この実施の形態1におけるファイル管理装置が使用するフラッシュメモリ5は32個のブロックを有しているものとする。

【0032】先に述べたように、ブロックとはフラッシュメモリの消去単位である。動作履歴記録領域10には20ブロック(0)が、FAT11にはブロック(1)が、FATCOPY12にはブロック(2)が、ディレクトリ領域13にはブロック(3)が、ディレクトリ待避領域14にはブロック(4)が、データ領域15にはブロック(5)～(30)が、データ待避領域16にはブロック(31)が割り当てられている。

【0033】データ領域15には1ブロック当たり128個のセクタが配置され、各セクタは0から3327までの番号で論理的にマッピングされている。FAT11はセクタ1つに対して2バイトのエントリを有し、初期状態(ファイルが何も存在しない状態)のエントリデータは‘0000’とする。1つのセクタに収まるファイルを記録した場合には、対応するセクタのFATエントリには、例えば‘FFFF’を記録する。ファイルが複数のセクタに記録される場合には、次に繋がるセクタの番号をFATエントリに記録する。ファイルの最終セクタには‘FFFF’を記録する。空きセクタの検索は、FAT11をスキャンし、エントリデータが‘0000’のものを検索することにより行うことができる。

【0034】図3はディレクトリ領域13のフォーマットを示す説明図であり、図において、20はディレクトリチェックバイトであり、ファイル登録済みのディレクトリであるか、未使用ディレクトリであるかを識別するためのデータが記録される。初期状態において、ディレクトリチェックバイト20は‘00’であるとする。ディレクトリチェックバイト20をスキャンすれば未使用ディレクトリを検索することができる。21はファイル名、22はファイル属性、23は日付、24は該当ファイルのデータが記録されている先頭のセクタの番号を示す開始セクタ番号、25はファイルサイズである。

【0035】動作履歴記録領域10は、ファイルの書込処理を実行するときの処理経過状況を示す唯一のデータ(以下、動作識別コードという)や、異常発生時にファイル管理装置を復旧するために必要となる情報を記録するためのエリアである。通常時は消去状態になっている。具体的な記録内容や動作履歴を記録するタイミングについては後述する。

【0036】なお、ファイル管理装置が使用するフラッシュメモリ5は、消去時に全てのビットが‘1’となる。ディレクトリ待避領域14は、ディレクトリ領域13の内容を書き換える際に使用するテンポラリ領域である。データ待避領域16は、データ領域15の内容を書き換える際に使用するテンポラリ領域である。FATCOPY12はFAT11と同じデータが記録される領域である。FATCOPY12を設ける手法はファイル管理装置の信頼性を向上するために従来からよく使われている。

【0037】次に動作について説明する。図4はファイル管理装置によるファイルの書込処理を示すフローチャートである。まず、MPU1は、キー入力部3からファイルの書込指令を受けると、そのファイルを保存するために必要なセクタ数を計算する。そして、MPU1は、FAT11をスキャンして、空きセクタを必要セクタ分検索する(ステップST1)。その際、必要セクタ分の空きセクタが無い場合、ファイルの書き込みが不可であることを通知する(ステップST17)。

【0038】MPU1は、空きセクタを必要セクタ分検索すると、動作履歴記録領域10に対して、ファイル書き込み中であることを示す動作識別コード‘A000’と、そのファイルを記録するセクタの番号を記録し(ステップST2)、そのファイルを該当セクタに記録する(ステップST3)。

【0039】そして、MPU1は、ディレクトリ領域13のディレクトリチェックバイト20をスキャンして、未使用ディレクトリを検索し(ステップST4)、ディレクトリ領域13の管理データをディレクトリ待避領域14にコピーする(ステップST5)。また、MPU1は、動作履歴記録領域10に対して、ディレクトリ更新中であることを示す動作識別コード‘A001’を追記する(ステップST6)。

【0040】MPU1は、動作識別コード‘A001’を動作履歴記録領域10に追記すると、ディレクトリ領域13をブロック消去する(ステップST7)。そして、MPU1は、新規ディレクトリデータと既存ディレクトリデータをマージして、ディレクトリ領域13にマージ後の管理データを記録する(ステップST8)。

【0041】次に、MPU1は、動作履歴記録領域10に対して、FAT更新中であることを示す動作識別コード‘A002’を追記する(ステップST9)。MPU1は、動作識別コード‘A002’を動作履歴記録領域

10に追記すると、FAT11をブロック消去する(ステップST10)。そして、MPU1は、新規FATエントリの記録と、既存FATエントリの書き戻しを実行する(ステップST11)。即ち、予め、FATCOPY12にコピーしていた既存FATエントリと、新規FATエントリをマージして、FATエントリを更新する。

【0042】次に、MPU1は、FAT11の更新を完了すると、動作履歴記録領域10に対して、FAT更新が完了したことを示す動作識別コード‘A003’を追記する(ステップST12)。MPU1は、動作識別コード‘A003’を動作履歴記録領域10に追記すると、FATCOPY12をブロック消去する(ステップST13)。

【0043】そして、MPU1は、次のファイル書込処理に備えるため、FAT11の管理データ(FATエントリ)をFATCOPY12にコピーする(ステップST14)。また、MPU1は、ディレクトリ待避領域14をブロック消去し(ステップST15)、動作履歴記録領域10をブロック消去する(ステップST16)。

【0044】ここで、上記ファイル書込処理の途中で電源が突然遮断されて異常終了した場合について考える。ここでは、一例として、図4のステップST8の処理中に電源が遮断されたものとして説明する。ただし、本発明の効果を明確にするため、最初に、復旧処理を全く実施しない場合について説明する。

【0045】再起動して、再度、ファイルの書込処理を試みると、次のような状況が発生する。まず、ステップST1において、FAT11をスキャンして空きセクタを検索するが、異常終了時点ではFATエントリへの書き込みが完了していないので、ファイル書込済みのセクタが空きセクタと認識され、ステップST3において、ファイルの上書き処理が実行される。

【0046】しかし、フラッシュメモリ5は、1度データを書き込んだ領域に対して、消去せずにデータの再書込を実行しても、正しくデータを記録することができない。したがって、シーケンス上は正常に書き込んだものとして処理が進行するが、誤った内容のファイルがフラッシュメモリ5に保存されることになる。

【0047】次に、動作履歴による復旧処理を実行する場合について説明する。図5は再起動時に実行する復旧処理を示すフローチャートである。まず、MPU1は、再起動時に動作履歴記録領域10のデータをチェックする(ステップST20)。

【0048】MPU1は、動作履歴記録領域10に動作識別コードの記録が残されていない場合(動作識別コードが消去されている場合)、異常終了状態が発生していないものと判断し、復旧処理を終了して通常の処理に移行する。一方、動作識別コードの記録が残されている場

合(動作識別コードが消去されていない場合)、異常終了状態が発生したものと判断し、ステップST21以降の処理を実行する。

【0049】MPU1は、異常終了状態の発生を認識すると、動作履歴記録領域10に最後に記録されている動作識別コードを解析し、ファイル書込処理のうち、どの処理を実行している時に異常終了したかを判断する(ステップST21)。この実施の形態1では、ステップST8の実行時に異常終了が発生したものと仮定しているので、動作履歴記録領域10の最後に記録されている動作識別コードは‘A001’のである。

【0050】異常終了の発生時期により、復旧処理の内容が若干異なるが、ステップST22以降に示す処理は動作識別コード‘A001’が検出された場合のフローチャートである。最後に記録されている動作識別コードが別なコードの場合は、ステップST29、ステップST30、ステップST31の処理に移行する。

【0051】MPU1は、ステップST21において、動作識別コード‘A001’が検出された場合、ディレクトリ領域13をブロック消去し(ステップST22)、ディレクトリ待避領域14の管理データをディレクトリ領域13にコピーする(ステップST23)。

【0052】また、MPU1は、動作履歴記録領域10に記録されているセクタ番号を参照して、そのセクタ番号が示すセクタを含むブロックのデータをデータ待避領域16にコピーする(ステップST24)。そして、MPU1は、そのセクタを含むブロックをブロック消去し(ステップST25)、そのセクタ以外の有効データを書き戻す処理を実行する(ステップST26)。即ち、そのブロックを構成するセクタのうち、そのセクタ番号が示すセクタ以外のセクタに格納されていたデータのみを元のブロックのセクタに書き戻す処理を実行する。

【0053】MPU1は、上記のようにしてデータ領域15の内容をデータの書込処理が実行される前の状態に戻すと、データ待避領域16をブロック消去し(ステップST27)、動作履歴記録領域10をブロック消去する(ステップST28)。上記復旧処理によりFATエントリが空きセクタと示しているセクタのデータが消去状態になり、また、中途半端に記録されたディレクトリの管理データも消去される。

【0054】以上のように、ファイルの書込処理中に動作識別コードを動作履歴記録領域10に逐次記録して、ファイルを記録するセクタの番号も動作履歴記録領域10に保存し、電源遮断などにより異常終了が発生すると、電源再起動時に動作識別コードを解析して、フラッシュメモリ5の内容を修復するようにしているので、突然の電源遮断などの突発的な異常終了状態に強く、データの信頼性が高いフラッシュメモリ5を記憶媒体に使用したファイル管理装置を得ることができるという効果を奏する。

【0055】図6はファイル管理装置によるファイルの削除処理を示すフローチャートである。まず、MPU1は、キー入力部3からファイルの削除指令を受けると、削除するファイルが格納されているセクタの番号を検索する(ステップST40)。そして、MPU1は、セクタ番号を検索すると、その削除するセクタを含むブロックのデータをデータ待避領域16にコピーする(ステップST41)。MPU1は、動作履歴記録領域10に対して、ファイル削除中であることを示す動作識別コード‘A100’と、そのファイルのファイル番号と、削除するセクタを含むブロックの番号とを記録する(ステップST42)。

【0056】そして、MPU1は、削除するセクタを含むブロックをブロック消去し(ステップST43)、削除するファイル以外の有効データを書き戻す処理を実行する(ステップST44)。即ち、そのブロックを構成するセクタのうち、削除するセクタ以外のセクタに格納されていたデータのみを元のブロックのセクタに書き戻す処理を実行する。

【0057】次に、MPU1は、データの書き戻し処理を実行すると、動作履歴記録領域10に対して、データ削除が完了したことを示す動作識別コード‘A101’を追記する(ステップST45)。MPU1は、動作識別コード‘A101’を動作履歴記録領域10に追記すると、ディレクトリ領域13の管理データをディレクトリ待避領域14にコピーする(ステップST46)。

【0058】次に、MPU1は、動作履歴記録領域10に対して、ディレクトリ更新中であることを示す動作識別コード‘A102’を追記する(ステップST47)。MPU1は、動作識別コード‘A102’を動作履歴記録領域10に追記すると、ディレクトリ領域13をブロック消去する(ステップST48)。また、MPU1は、削除したファイルのディレクトリを未使用状態にして、最新の管理データをディレクトリ領域13に記録する(ステップST49)。なお、削除対象ファイル以外のファイルに係る管理データは、ディレクトリ待避領域14にコピーした管理データをディレクトリ領域13にコピーする。

【0059】次に、MPU1は、動作履歴記録領域10に対して、FAT更新中であることを示す動作識別コード‘A103’を追記する(ステップST50)。MPU1は、動作識別コード‘A103’を動作履歴記録領域10に追記すると、FAT11をブロック消去する(ステップST51)。また、MPU1は、新規FATエントリの記録と、既存FATエントリの書き戻しを実行する(ステップST52)。即ち、予め、FATCOPY12にコピーしていた既存FATエントリと、新規FATエントリをマージして、FATエントリを更新する。

【0060】次に、MPU1は、FAT11の更新を完

了すると、動作履歴記録領域10に対して、FAT更新が完了したことを示す動作識別コード‘A104’を追記する(ステップST53)。MPU1は、動作識別コード‘A104’を動作履歴記録領域10に追記すると、FATCOPY12をブロック消去する(ステップST54)。

【0061】そして、MPU1は、今回のファイル削除処理に備えるため、FAT11の管理データ(FATエントリ)をFATCOPY12にコピーする(ステップST55)。また、MPU1は、データ待避領域16をブロック消去するとともに(ステップST56)、ディレクトリ待避領域14をブロック消去し(ステップST57)、動作履歴記録領域10をブロック消去する(ステップST58)。

【0062】ここで、上記ファイル削除処理の途中で電源が突然遮断されて異常終了した場合について考える。ここでは、一例として、図6のステップST43の処理後に電源が遮断されたものとして説明する。なお、削除対象のセクタを含むブロックに格納されていたファイルは全て消去されているので、復旧処理を全く実施せずに再起動した場合、そのブロックに格納されていたファイルの読み出しを試みると、消去状態のファイルを読み出すことになる。

【0063】次に、動作履歴による復旧処理を実行する場合について説明する。図7は再起動時に実行する復旧処理を示すフローチャートである。まず、MPU1は、再起動時に動作履歴記録領域10のデータをチェックする(ステップST60)。

【0064】MPU1は、動作履歴記録領域10に動作識別コードの記録が残されていない場合(動作識別コードが消去されている場合)、異常終了状態が発生していないものと判断し、復旧処理を終了して通常の処理に移行する。一方、動作識別コードの記録が残されている場合(動作識別コードが消去されていない場合)、異常終了状態が発生したものと判断し、ステップST61以降の処理を実行する。

【0065】MPU1は、異常終了状態の発生を認識すると、動作履歴記録領域10に最後に記録されている動作識別コードを解析し、ファイル削除処理のうち、どの処理を実行している時に異常終了したかを判断する(ステップST61)。この実施の形態1では、ステップST43の実行後に異常終了が発生したものと仮定しているので、動作履歴記録領域10の最後に記録されている動作識別コードは‘A100’である。

【0066】異常終了の発生時期により、復旧処理の内容が若干異なるが、ステップST62以降に示す処理は動作識別コード‘A100’が検出された場合のフローチャートである。最後に記録されている動作識別コードが別なコードの場合は、ステップST67、ステップST68、ステップST69、ステップST70の処理に

移行する。

【0067】MPU1は、ステップST61において、動作識別コード‘A100’が検出された場合、動作履歴記録領域10に記録されているセクタ番号を参照して、そのセクタ番号が示すセクタを含むブロックをブロック消去する(ステップST62)。そして、MPU1は、図6のステップST41において、データ待避領域16にコピーされたデータを上記のブロックにコピーする(ステップST63)。

【0068】MPU1は、上記のようにしてデータ領域15の内容をデータの削除処理が実行される前の状態に戻すと、データ待避領域16をブロック消去するとともに(ステップST64)、動作履歴記録領域10をブロック消去し(ステップST65)、ファイル削除処理を再実行する(ステップST66)。なお、削除するファイルの番号は動作履歴記録領域10に記録されているので、動作履歴記録領域10を参照することにより認識することができる。上記復旧処理により、消去された有効なファイルが復活する。

【0069】以上のように、この実施の形態1によれば、ファイル削除処理中に動作識別コードを動作履歴記録領域10に逐次記録して、削除するファイルの番号と削除するファイルが格納されているセクタを含むブロックの番号も動作履歴記録領域10に保存し、電源遮断などにより異常終了が発生すると、電源再起動時に動作識別コードを解析して、フラッシュメモリ5の内容を修復するようにしているので、突然の電源遮断などの突発的な異常終了状態に強く、データの信頼性が高いフラッシュメモリ5を記憶媒体に使用したファイル管理装置を得ることができる効果を奏する。

【0070】実施の形態2。上記実施の形態1では、ファイル書込処理又はファイル削除処理が正常終了した時に動作履歴記録領域10をブロック消去するものについて示したが、ファイル書込処理又はファイル削除処理が正常終了したことを示す動作識別コードを動作履歴記録領域10に追記するようにして、動作履歴記録領域10の空き領域が一定サイズ以下になったら、動作履歴記録領域10をブロック消去するようにしてもよい。なお、ファイル管理装置の構成及びフラッシュメモリ5上の論理フォーマットは、上記実施の形態1における図1及び図2と同様である。

【0071】次に動作について説明する。図8はファイル管理装置によるファイルの書込処理を示すフローチャートである。まず、MPU1は、キー入力部3からファイルの書込指令を受けると、そのファイルを保存するために必要なセクタ数を計算する。そして、MPU1は、FAT11をスキャンして、空きセクタを必要セクタ分検索する(ステップST80)。その際、必要セクタ分の空きセクタが無い場合、ファイルの書き込みが不可であることを通知する(ステップST98)。

【0072】MPU1は、空きセクタを必要セクタ分検索すると、動作履歴記録領域10の空き領域をチェックする(ステップST81)。そして、MPU1は、空き領域が一定サイズ以下の場合は、動作履歴記録領域10をブロック消去する(ステップST82)。一方、空き領域が一定サイズ以上の場合は、動作履歴記録領域10に対して、ファイル書き込み中であることを示す動作識別コード‘A000’と、そのファイルを記録するセクタの番号を記録し(ステップST83)、そのファイルを該当セクタに記録する(ステップST84)。

【0073】そして、MPU1は、ディレクトリ領域13のディレクトリチェックバイト20をスキャンして、未使用ディレクトリを検索し(ステップST85)、ディレクトリ領域13の管理データをディレクトリ待避領域14にコピーする(ステップST86)。また、MPU1は、動作履歴記録領域10に対して、ディレクトリ更新中であることを示す動作識別コード‘A001’を追記する(ステップST87)。

【0074】MPU1は、動作識別コード‘A001’を動作履歴記録領域10に追記すると、ディレクトリ領域13をブロック消去する(ステップST88)。そして、MPU1は、新規ディレクトリデータと既存ディレクトリデータをマージして、ディレクトリ領域13にマージ後の管理データを記録する(ステップST89)。

【0075】次に、MPU1は、動作履歴記録領域10に対して、FAT更新中であることを示す動作識別コード‘A002’を追記する(ステップST90)。MPU1は、動作識別コード‘A002’を動作履歴記録領域10に追記すると、FAT11をブロック消去する(ステップST91)。そして、MPU1は、新規FATエントリの記録と、既存FATエントリの書き戻しを実行する(ステップST92)。即ち、予め、FATCOPY12にコピーしていた既存FATエントリと、新規FATエントリをマージして、FATエントリを更新する。

【0076】次に、MPU1は、FAT11の更新を完了すると、動作履歴記録領域10に対して、FAT更新が完了したことを示す動作識別コード‘A003’を追記する(ステップST93)。MPU1は、動作識別コード‘A003’を動作履歴記録領域10に追記すると、FATCOPY12をブロック消去する(ステップST94)。

【0077】そして、MPU1は、次のファイル書込処理に備えるため、FAT11の管理データ(FATエントリ)をFATCOPY12にコピーする(ステップST95)。また、MPU1は、ディレクトリ待避領域14をブロック消去し(ステップST96)、動作履歴記録領域10に対して、ファイル書込処理が正常終了したことを示す動作識別コード‘A0FF’を追記する(ステップST97)。

【0078】上記実施の形態1では、電源起動時に動作履歴記録領域10が消去されていない場合、動作識別コードを解析して復旧処理を実行していたが、この実施の形態2では、動作履歴記録領域10に最後に記録された動作識別コードが、正常終了したことを示す動作識別コードで無い場合、その動作識別コードを解析して、上記実施の形態1と同様の復旧処理を実行するようにする。

【0079】ファイル削除処理についても、上記ファイル書込処理と同様に、動作履歴記録領域10に最後に記録された動作識別コードが、正常終了したことを示す動作識別コードで無い場合、その動作識別コードを解析して、上記実施の形態1と同様の復旧処理を実行するようにする。

【0080】フラッシュメモリ5は、データの読込時間と比べて、データの書き込みや、ブロック消去に要する時間が長いという欠点を有している。以上のように、この実施の形態2によれば、ファイル書込処理あるいはファイル削除処理が正常終了した時には、正常終了したことを示す動作識別コードを追記するようにし、動作履歴記録領域10の空き領域が一定サイズ以下になったらブロック消去するようにしているため、消去時間のオーバーヘッドを削減することができるようになり、その結果、システムのスループットが向上する効果を奏する。

【0081】実施の形態3。上記実施の形態1、2では、ファイル書込処理又はファイル削除処理中に動作識別コードを動作履歴記録領域10に逐次記録するものについて示したが、図12に示すように、動作識別コードを逐次記録する際、その動作識別コードを動作履歴記録領域10に対して、2回続けて記録するようにしてもよい。なお、ファイル管理装置の構成及びフラッシュメモリ5上の論理フォーマットは、上記実施の形態1における図1及び図2と同様である。

【0082】上記実施の形態1では、ファイル書込処理中における異常終了の例として、図4のステップST8の処理を実行している時に、電源が遮断された場合について説明した。しかし、稀なケースとして動作識別コードを記録している最中に電源が遮断される可能性もある。動作履歴記録中に電源が遮断されると動作識別コードが正しく記録されず、復旧処理が正しく実行されない可能性がある。

【0083】図9はセクタ番号が‘0010’、‘0011’、‘0012’、‘0013’、‘0014’のセクタにファイルを書き込む場合において、ステップST3を実行する前の動作履歴の記録状態を示す説明図である。‘A000’の後に続く‘0010’、‘0011’、‘0012’、‘0013’、‘0014’はファイルを記録するセクタの番号である。‘FFFF’はデータが消去状態であることを示している。

【0084】一方、図10はステップST2において、ファイルを記録するセクタの番号を動作履歴記録領域1

0に記録している最中に異常終了した場合における動作履歴の記録状態の一例を示したものである。本来、‘0013’と記録されるべきところのデータが誤って‘0000’と記録されてしまった状態を示している。この場合、検出される動作識別コードは、‘A000’であるので、図5の復旧処理①(ステップST29)が実行される。図11は復旧処理①の処理内容を示すフローチャートである。

【0085】MPU1は、動作履歴記録領域10に記録されているセクタ番号を参照して、そのセクタ番号が示すセクタを含むブロックのデータをデータ待避領域16にコピーする(ステップST100)。そして、MPU1は、そのセクタを含むブロックをブロック消去(ステップST101)、そのセクタ以外の有効データを書き戻す処理を実行する(ステップST102)。即ち、そのブロックを構成するセクタのうち、そのセクタ番号が示すセクタ以外のセクタに格納されていたデータのみを元のブロックのセクタに書き戻す処理を実行する。そして、MPU1は、データ待避領域16をブロック消去(ステップST103)、動作履歴記録領域10をブロック消去する(ステップST104)。

【0086】以上のように、復旧処理は、動作履歴記録領域10に記録されているセクタ番号のセクタに格納されているデータを消去するための処理を実行するので、図10に示した動作履歴記録状態で復旧処理を実行すると、セクタ番号‘0000’のセクタに格納されているデータが誤って消去されることになる。

【0087】図12は図10と同様に、ステップST2において、ファイルを記録するセクタの番号を動作履歴記録領域10に記録している最中に異常終了した場合における動作履歴の記録状態の一例を示したものである。本来、‘0013’が2回続けて記録されるべきところのデータが‘0000’、‘FFFF’となっている。

【0088】この実施の形態3では、図11のステップST100において、同じ値が2回続けて記録されている動作識別コードのみを有効データとして扱う。したがって、‘0000’は無効なデータとして扱われ、セクタ番号‘0000’のデータが誤って消去されることは無い。

【0089】ここでは、ファイル書込処理の動作について説明したが、ファイル削除処理においても、動作履歴に記録するデータが異なるだけで、動作は同様であるので説明を省略する。

【0090】以上のように、この実施の形態3によれば、ファイル書込処理時においてはファイルを書き込む論理アドレスと処理経過状況を識別する動作識別コードを2回続けて動作履歴記録領域10に逐次記録し、ファイル削除処理時においては削除するファイルの番号と、消去中のブロックの番号と、処理経過状況を識別する動作識別コードを2回続けて動作履歴記録領域10に逐次

記録するようにしているので、動作履歴記録中に異常終了しても、正しく動作履歴を解析して、復旧処理を実行することができるようになり、その結果、データの信頼性が更に向上したフラッシュメモリ5を記憶媒体に使用したファイル管理装置を得ることができる効果を奏する。

【0091】実施の形態4. この実施の形態4のファイル管理装置は、フラッシュメモリ5のハードウェア・ステータスのエラーを検出するようにし、エラーを検出した場合には、ファイル書込処理又はファイル削除処理を中断して、復旧処理を実行するようにしたものである。

【0092】ファイル管理装置の構成及びフラッシュメモリ5上の論理フォーマットは、上記実施の形態1における図1及び図2と同様である。ただし、この実施の形態4では、MPU1は、更に検出手段及び制御手段を構成する。

【0093】フラッシュメモリ5は、DRAMやSRAMと異なり、データの書き込みやブロックの消去は、MPU1からソフトウェア的なコマンドを発行することにより実行される。図13はフラッシュメモリ5に対するデータの書込処理を実行するプログラム・コマンドを示すフローチャートである。

【0094】まず、MPU1は、フラッシュメモリ5に対してプログラムセットアップコマンドを書き込む処理を実行する(ステップST110)。そして、MPU1は、プログラムセットアップコマンドを書き込むと、フラッシュメモリ5のプログラムアドレスに対してデータを書き込む処理を実行する(ステップST111)。

【0095】MPU1は、フラッシュメモリ5のステータスがレディ状態になるまで待機し(ステップST112)、フラッシュメモリ5のステータスがレディ状態になると、フラッシュメモリ5のエラーステータスをチェックする(ステップST113)。これにより、MPU1は、フラッシュメモリ5のハードウェア的なエラーを検出することができる。したがって、フラッシュメモリ5のハードウェア的なエラーが検出された場合には、以降のファイル書込処理を中断し、動作履歴による復旧処理に移行する。

【0096】図14はフラッシュメモリ5に対するブロック消去を実行するブロックイレース・コマンドを示すフローチャートである。まず、MPU1は、フラッシュメモリ5に対してブロックイレース・セットアップコマンドを書き込む処理を実行する(ステップST120)。そして、MPU1は、ブロックイレース・セットアップコマンドを書き込むと、フラッシュメモリ5に対してブロックイレース・確認コマンドを書き込む処理を実行する(ステップST121)。

【0097】MPU1は、フラッシュメモリ5のステータスがレディ状態になるまで待機し(ステップST122)、フラッシュメモリ5のステータスがレディ状態に

なると、フラッシュメモリ5のエラーステータスをチェックする(ステップST123)。これにより、MPU1は、フラッシュメモリ5のハードウェア的なエラーを検出することができる。したがって、フラッシュメモリ5のハードウェア的なエラーが検出された場合には、以降のファイル削除処理を中断し、動作履歴による復旧処理に移行する。

【0098】以上のように、この実施の形態4によれば、フラッシュメモリ5のハードウェア・ステータスのエラーを検出すると、ファイル書込処理あるいはファイル削除処理を中断して、復旧処理を実行するようにしているので、データの信頼性が更に向上したフラッシュメモリ5を記憶媒体に使用したファイル管理装置を得ることができる効果を奏する。

【0099】実施の形態5. 上記実施の形態4では、フラッシュメモリ5のハードウェア・ステータスのエラーを検出すると、ファイル書込処理あるいはファイル削除処理を中断して、復旧処理を実行するものについて示したが、フラッシュメモリ5にコマンドを発行した時に計測を開始するタイマを設け、フラッシュメモリ5から一定時間応答がない場合は、ファイル書込処理あるいはファイル削除処理を中断し、復旧処理を実行するようにしてもよい。ファイル管理装置の構成及びフラッシュメモリ5上の論理フォーマットは、上記実施の形態1における図1および図2と同様である。

【0100】図13及び図14におけるステップST112、ST122において、ステータスがレディ状態になるまで待機する処理は、フラッシュメモリ5から読み出されるステータスをポーリングし、そのステータス中の特定のビットをチェックしている。何らかの不具合により、その特定のビットがレディ状態にならないと、プログラムがポーリング処理から次の処理に移行できない場合がある。

【0101】図15はこの実施の形態5におけるプログラム・コマンドの処理内容を示すフローチャートである。まず、MPU1は、フラッシュメモリ5に対してプログラムセットアップコマンドを書き込む処理を実行する(ステップST130)。そして、MPU1は、プログラムセットアップコマンドを書き込むと、フラッシュメモリ5のプログラムアドレスに対してデータを書き込む処理を実行する(ステップST131)。その際、MPU1は、タイマを起動する(ステップST132)。

【0102】MPU1は、上記実施の形態4と同様に、フラッシュメモリ5のステータスがレディ状態になるまで待機するが(ステップST133)、その待機中、タイマの値をチェックし(ステップST134)、タイマの値が一定値以上になると、フラッシュメモリ5のステータスが非レディ状態のままでも、ファイル書込処理を中断して、復旧処理を実行するようにする。

【0103】ただし、タイマの値が一定値以下である限

10

20

30

40

50

り、待機状態を継続し、フラッシュメモリ5のステータスがレディ状態になると、上記実施の形態4と同様に、フラッシュメモリ5のエラーステータスをチェックする(ステップST135)。エラーが検出された場合には、ファイル書込処理を中断して、復旧処理を実行する。

【0104】この実施の形態5では、ファイル書込処理中にタイマを起動して、タイマの値が一定値以上になると、復旧処理を実行するものについて示したが、ファイル削除処理では、図14におけるステップST121の処理後にタイマを起動するようにすればよく、ファイル書込処理と同様に復旧処理を開始することができる。

【0105】以上のように、この実施の形態5によれば、フラッシュメモリ5にコマンドを発行した時に計測を開始するタイマを設け、フラッシュメモリ5から一定時間応答がない場合は、ファイル書込処理あるいはファイル削除処理を中断して、復旧処理を実行するようにしているので、データの信頼性が更に向上したフラッシュメモリ5を記憶媒体に使用したファイル管理装置を得ることができる効果を奏する。

【0106】実施の形態6。近年のフラッシュメモリの中には、2個のバンクから構成され、一方のバンクに対する書込処理あるいは削除処理中に、他方のバンクからデータを読み出すことが可能なものがある。上記機能は、デュアルオペレーション機能とか、バックグラウンド・オペレーション機能などと呼ばれている。この実施の形態6のファイル管理装置は、上記機能を有したフラッシュメモリ5を使用し、ファイル書込処理やファイル削除処理における処理時間のパフォーマンス向上を図るものである。

【0107】図16はファイル管理装置におけるフラッシュメモリ5上の論理フォーマットを示す説明図である。フラッシュメモリ5は、バンク(1)とバンク(2)に区分けされ、バンク(1)にはブロック(0)～(15)が、バンク(2)にはブロック(16)～(31)が割り当てられている。バンク(1)に対して書込処理や削除処理を実行している時に、バンク(2)のデータを読み出すことが可能であり、逆に、バンク(2)に対して書込処理や削除処理を実行している時に、バンク(1)のデータを読み出すことが可能である。

【0108】この実施の形態6では、動作履歴記録領域10にはブロック(0)が、FAT11にはブロック(1)が、FATCOPY12にはブロック(29)が、ディレクトリ領域13にはブロック(2)が、ディレクトリ待避領域14にはブロック(30)が、データ領域15にはブロック(3)～(28)が、データ待避領域16にはブロック(31)が割り当てられている。ファイル管理装置の構成は実施の形態1における図1と同様である。ただし、このこの実施の形態6では、MP

U1は更に読込手段を構成している。

【0109】前の実施の形態で述べたように、ファイル書込処理、ファイル削除処理、復旧処理等においては、FAT、ディレクトリ、ファイル等を、任意のブロックから別なブロックにコピーする処理が頻繁に実行される。図17はデュアルオペレーション機能を有していないフラッシュメモリを使用した場合におけるコピー処理実行時の動作を説明する説明図である。

【0110】コピー元のブロックから一定サイズのデータを読み出して、RAM8やMPU1内部のRAMにデータを格納し、次にフラッシュメモリ5にプログラム・コマンドを発行して、別なブロックにデータ書き込む。上記読出動作と、書込動作を繰り返し実行して、1ブロック分のデータをコピーする。書込時間は読出時間に比べて長い時間を要するが、MPU1は書込動作が終了したことを示すステータスが得られるまで待機してから、次のデータの読出動作に移行する。

【0111】図18はデュアルオペレーション機能を有しているフラッシュメモリを使用した場合におけるコピー処理実行時の動作を説明する説明図である。FAT11、ディレクトリ領域13及びデータ領域15の一部のブロックは、バンク(1)に構成され、FATCOPY12、ディレクトリ待避領域14及びデータ待避領域16はバンク(2)に構成されている。

【0112】したがって、バンク(1)のデータをバンク(2)に、あるいは、逆にバンク(2)のデータをバンク(1)にコピーする場合、図18に示したように、読出動作と書込動作をオーバーラップして実行することができるため、データ処理時間を短縮することができる。以上、ブロック間でのデータコピー処理を例に、デュアルオペレーション機能によりデータ処理時間が短縮されることを説明したが、データコピー処理以外にも、ファイル書込と未使用ディレクトリの検索を平行して行うなど、別な処理に適用することも可能である。

【0113】以上のように、この実施の形態6によれば、デュアルオペレーション機能を有したフラッシュメモリ5を使用し、データのコピーなどが頻繁に行われるブロックは別々のバンクになるように論理フォーマットを構成しているので、データ処理時間が短縮され、システムのスループットが向上する効果を奏する。

【0114】

【発明の効果】以上のように、この発明によれば、書込手段が書込処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリ5の動作履歴記録領域に記録し、その書込処理又は更新処理が正常終了すると、その識別コードを消去する動作履歴記録手段を設け、そのフラッシュメモリ5の動作履歴記録領域に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するように構成したので、書込手段が書込処

理又は更新処理を実行している時に電源遮断などの異常状態が発生しても、ファイルやデータが破壊されことなく、ファイル操作を正常に再開することができる効果がある。

【0115】この発明によれば、書込手段が書込処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その書込処理又は更新処理が正常終了すると、正常終了を示す識別コードを動作履歴記録領域に記録する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に正常終了を示す識別コードが記録されていない場合、その処理内容を示す識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するように構成したので、書込手段が書込処理又は更新処理を実行している時に電源遮断などの異常状態が発生しても、ファイルやデータが破壊されことなく、ファイル操作を正常に再開することができる効果がある。

【0116】この発明によれば、識別コードの解析結果に応じた復旧処理を実行して、データ領域及び管理領域に格納されているデータを書込処理又は更新処理が実行される前の内容に戻すように構成したので、突然の電源遮断などの突発的な異常終了状態に強く、データの信頼性が高いフラッシュメモリが得られる効果がある。

【0117】この発明によれば、削除手段が削除処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その削除処理又は更新処理が正常終了すると、その識別コードを消去する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に識別コードの記録が残されている場合、その識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するように構成したので、削除手段が削除処理又は更新処理を実行している時に電源遮断などの異常状態が発生しても、ファイルやデータが破壊されことなく、ファイル操作を正常に再開することができる効果がある。

【0118】この発明によれば、削除手段が削除処理又は更新処理を実行する際、その処理内容を示す識別コードをフラッシュメモリの動作履歴記録領域に記録し、その削除処理又は更新処理が正常終了すると、正常終了を示す識別コードを動作履歴記録領域に記録する動作履歴記録手段を設け、そのフラッシュメモリの動作履歴記録領域に正常終了を示す識別コードが記録されていない場合、その処理内容を示す識別コードを解析して、そのデータ領域及び管理領域に格納されているデータを復旧するように構成したので、削除手段が削除処理又は更新処理を実行している時に電源遮断などの異常状態が発生しても、ファイルやデータが破壊されことなく、ファイル操作を正常に再開することができる効果がある。

【0119】この発明によれば、識別コードの解析結果に応じた復旧処理を実行して、データ領域及び管理領域

に格納されているデータを削除処理又は更新処理が実行される前の内容に戻すように構成したので、突然の電源遮断などの突発的な異常終了状態に強く、データの信頼性が高いフラッシュメモリが得られる効果がある。

【0120】この発明によれば、同一の識別コードが2回続けて記録されている場合に限り、その識別コードを有効な識別コードであるとして取り扱うように構成したので、動作履歴記録中に異常終了しても、正しく動作履歴を解析して、復旧処理を実行することができるようになり、その結果、データの信頼性が更に向上したフラッシュメモリを得ることができる効果がある。

【0121】この発明によれば、検出手段がハードウェアエラーを検出すると、書込手段の処理を中止させて復旧手段の処理を開始させるように構成したので、データの信頼性が更に向上したフラッシュメモリを得ることができる効果がある。

【0122】この発明によれば、検出手段がハードウェアエラーを検出すると、削除手段の処理を中止させて復旧手段の処理を開始させるように構成したので、データの信頼性が更に向上したフラッシュメモリを得ることができる効果がある。

【0123】この発明によれば、検出手段がフラッシュメモリからハードウェアエラーのステータス信号を一定時間以上受信できない場合、その検出手段がハードウェアエラーを検出した場合と同様の処理を実行するように構成したので、フラッシュメモリに何らかの不具合が発生して、ハードウェアエラーのステータス信号を受信できない場合でも、復旧手段が復旧処理を実行することができるようになり、その結果、データの信頼性が更に向上したフラッシュメモリを得ることができる効果がある。

【0124】この発明によれば、書込手段が一方のバンクに対してファイルデータを書き込んでいる時、他方のバンクからデータを読み込む読込手段を設けるように構成したので、データ処理時間が短縮され、システムのスループットが向上する効果がある。

【0125】この発明によれば、削除手段が一方のバンクからセクタを削除している時、他方のバンクからデータを読み込む読込手段を設けるように構成したので、データ処理時間が短縮され、システムのスループットが向上する効果がある。

【図面の簡単な説明】

【図1】 この発明の実施の形態1によるファイル管理装置を示す構成図である。

【図2】 ファイル管理装置におけるフラッシュメモリ5上の論理フォーマットを示す説明図である。

【図3】 ディレクトリ領域13のフォーマットを示す説明図である。

【図4】 ファイル管理装置によるファイルの書込処理を示すフローチャートである。

【図5】 再起動時に実行する復旧処理を示すフローチャートである。

【図6】 ファイル管理装置によるファイルの削除処理を示すフローチャートである。

【図7】 再起動時に実行する復旧処理を示すフローチャートである。

【図8】 ファイル管理装置によるファイルの書込処理を示すフローチャートである。

【図9】 ステップST3を実行する前の動作履歴の記録状態を示す説明図である。

【図 10】 ファイルを記録するセクタの番号を動作履歴記録領域 10 に記録している最中に異常終了した場合における動作履歴の記録状態の一例を示す説明図である。

【図11】 復旧処理①の処理内容を示すフローチャートである。

【図 12】 ファイルを記録するセクタの番号を動作履歴記録領域 10 に記録している最中に異常終了した場合における動作履歴の記録状態の一例を示す説明図である。

【図 13】 フラッシュメモリ 5 に対するデータの書込処理を実行するプログラム・コマンドを示すフローチャートである。

【図 14】 フラッシュメモリ 5 に対するブロック消去
を実行するブロックイレーズ・コマンドを示すフローチ
ャートである。

*【図15】 プログラム・コマンドの処理内容を示すフローチャートである。

【図16】 ファイル管理装置におけるフラッシュメモリ5上の論理フォーマットを示す説明図である。

【図17】デュアルオペレーション機能を有していないフラッシュメモリを使用した場合におけるコピー処理実行時の動作を説明する説明図である。

【図18】デュアルオペレーション機能を有しているフラッシュメモリを使用した場合におけるコピー処理実行時の動作を説明する説明図である。

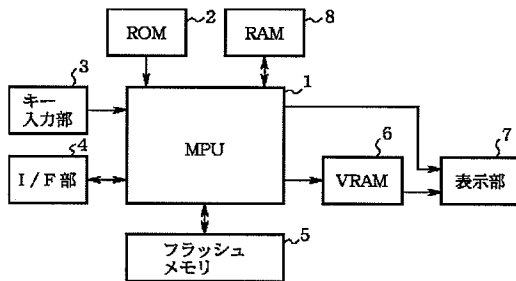
【図 19】 従来のファイル管理装置におけるフラッシュメモリ上の論理フォーマットを示す説明図である。

【図20】 従来のファイル管理装置の処理内容を示すフローチャートである。

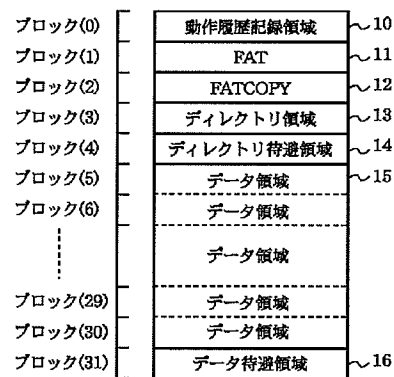
【符号の説明】

1 MPU（書込手段、削除手段、動作履歴記録手段、
復旧手段、検出手段、制御手段、読込手段）、2 ROM、3 キー入力部、4 I/F部、5 フラッシュメモ
リ、6 VRAM、7 表示部、8 RAM、10
20 動作履歴記録領域、11 FAT（管理領域）、12
FATCOPY、13 ディレクトリ領域（管理領
域）、14 ディレクトリ待避領域、15 データ領
域、16 データ待避領域、20 ディレクトリチェッ
クバイト、21 ファイル名、22 ファイル属性、2
3 日付、24 開始セクタ番号、25 ファイルサイ
ズ。

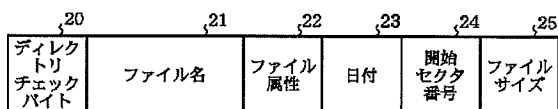
【図 1】



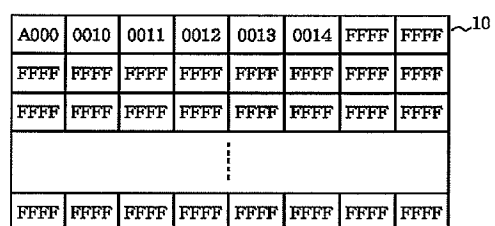
【図 2】



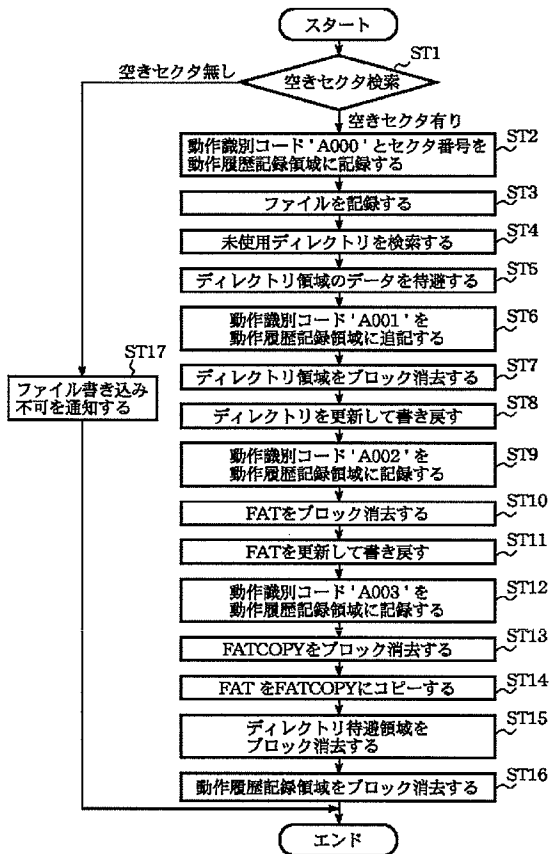
【图 3】



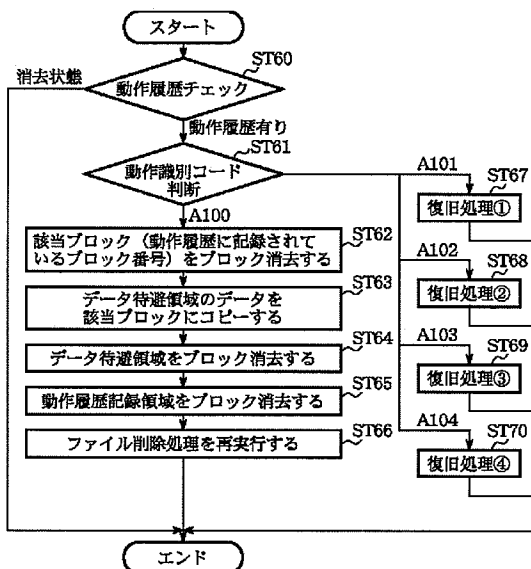
【图9】



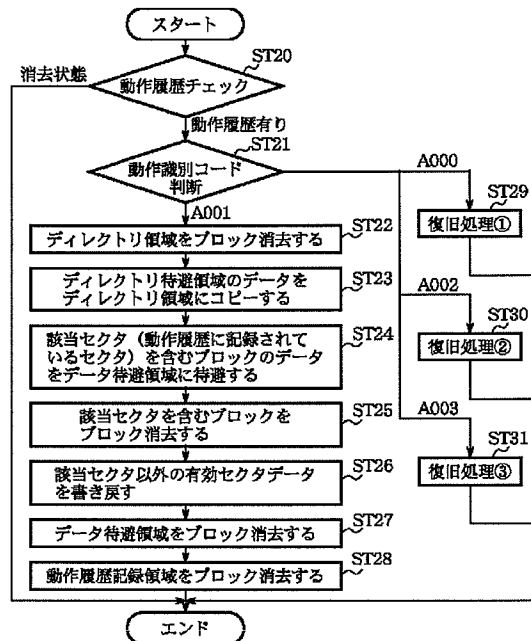
【図4】



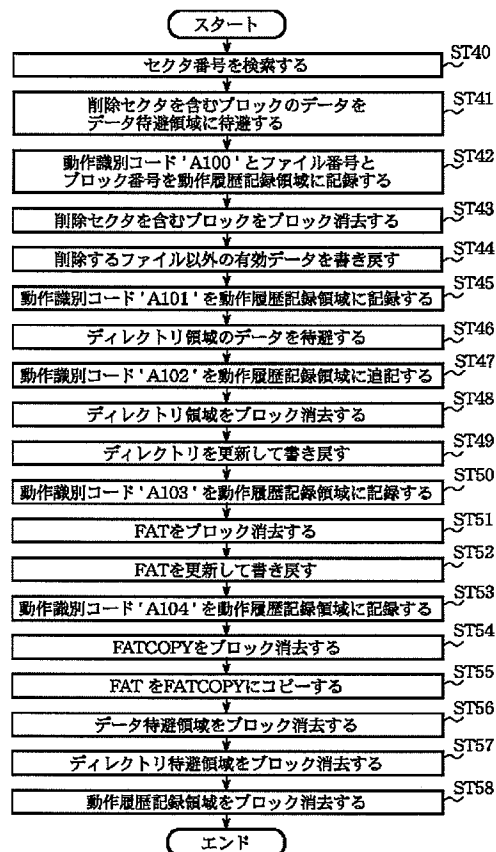
【図7】



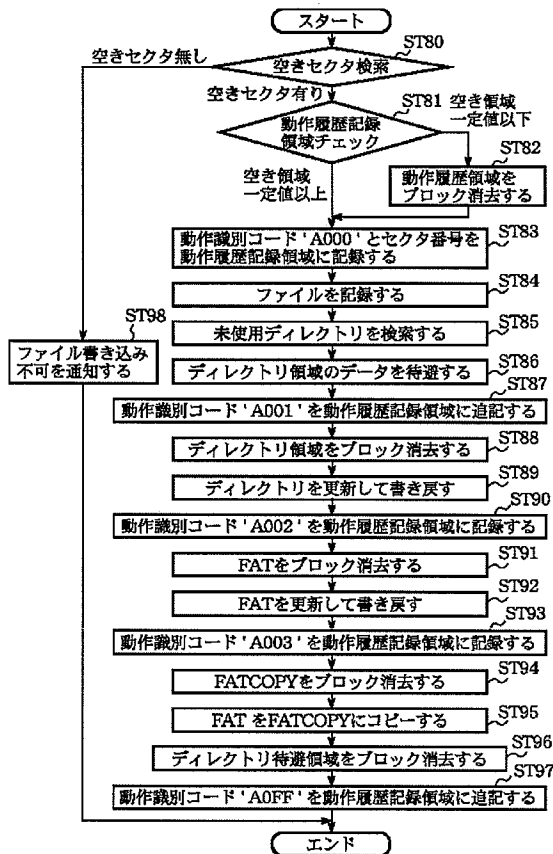
【図5】



【図6】



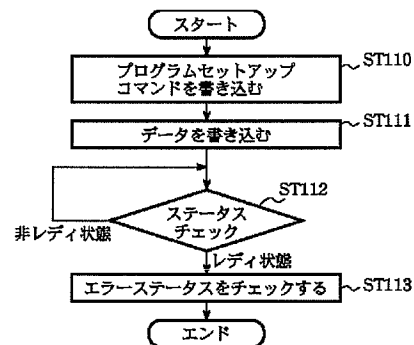
【図8】



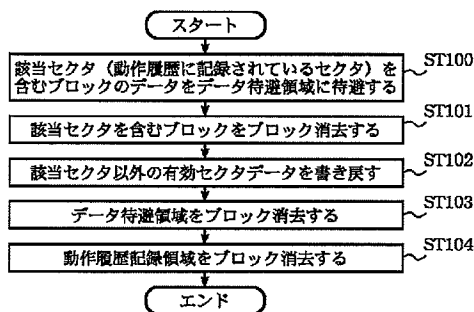
【図10】

A000	0010	0011	0012	0000	FFFF	FFFF	FFFF	~10
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	

【図13】



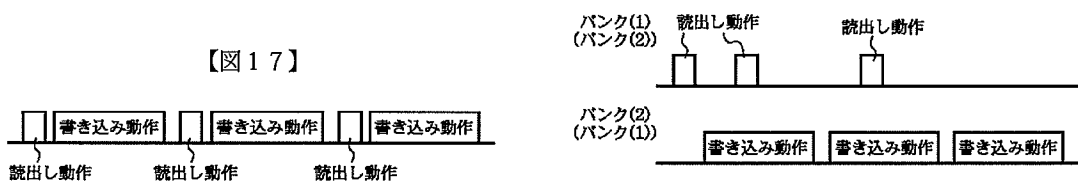
【図11】



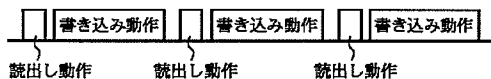
【図12】

A000	A000	0010	0010	0011	0011	0012	0012	~10
0000	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	

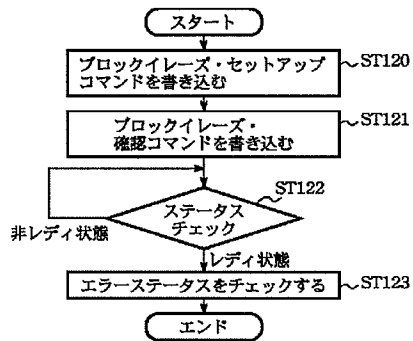
【図18】



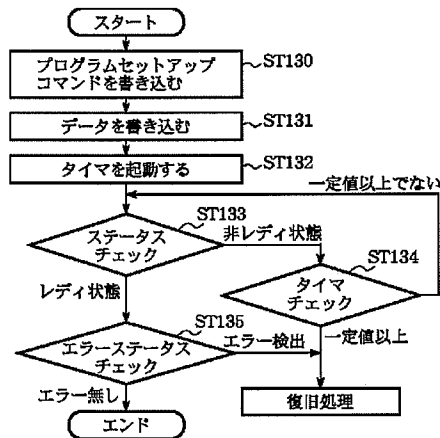
【図17】



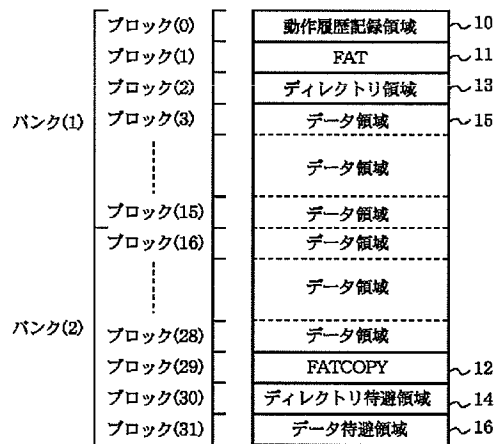
【図14】



【図15】



【図16】



【図19】



【図20】

